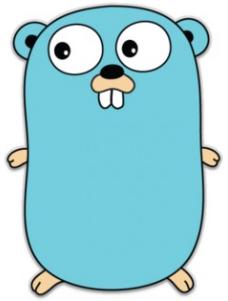
Basics of Golang

By

Prince Khanna

Agenda



Go

- What is Golang?
- How is it useful?
- Basic Concept of Golang
- Demo
- Q&A

What?

Go is an **open-source** programming language focused on **simplicity**, **reliability**, and **efficiency**.

Go programming language is a **statically-typed language** with syntax like that of C. It provides **garbage collection**, **type safety**, **dynamic-typing capability**, many advanced built-in types such as variable length arrays and key-value maps. It also provides a rich standard library.

Benefits of Golang

- **Golang is Fast**
- **Golang is Easy To Learn**
- **Golang is Well-Scaled**
- **Comprehensive Programming Tools**
- But it does not support generic functions yet.

Basic Concepts of Golang

- Packages
- Imports
- Functions
- Variables
- For Loop
- Arrays
- Slice
- Structure
- Creating server in Go
- Rest APIs in Go

Package

Every Go program is made up of packages. Programs start running in package main.

test.go

```go
package main

import "fmt"

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

Imports

There are 2 ways to write imports

```
import "fmt"
import "math"
```

```
import(
    "fmt"
    "math"
)
```

Functions

The general form of a function definition in Go programming language is as follows:

```
func function_name( [parameter list] ) [return_types]
{
   body of the function
}
```

A function can take zero or more arguments. In example below , add takes two parameters of type int. Notice that the type comes after the variable name.

function.go

```
1   package main
2
3   import "fmt"
4
5   func add(a int, b int) int {
6       return a + b
7   }
8
9   func main() {
10      fmt.Println("sum is ",add(10, 19))
11  }
```

Variables

A variable definition means to tell the compiler where and how much to create the storage for the variable. A variable definition specifies a data type and also contains a list of one or more variables of that type as follows:

```
var variable_list optional_data_type;
```

The var statement declares a list of variables; as in function argument lists, the type is last. A var statement can be at package or function level. We see both in the example below.

variables.go

```
1   package main
2
3   import "fmt"
4
5   var node, golang, angular bool
6
7   func main() {
8       var x int
9       fmt.Println(x, node, golang, angular)
10  }
```

For Loop

A for loop is a repetition control structure that will allow you to efficiently write a loop that needs to execute a specific number of times.

The syntax of a for loop in Go programming language is:

```
for [condition |  ( init; condition; increment ) | Range]
{
   statement(s);
}
```

Note: Unlike other languages like C, Java, or Javascript there are no parentheses surrounding the three components of the for statement and the braces { } are always required.

for.go

```go
package main

import "fmt"

func main() {
    sum := 0
    for i := 0; i < 8; i++ {
        sum += i
    }
    fmt.Println("Total sum =",sum)
}
```

Arrays

To declare an array in Go, a programmer specifies the type of the elements and the number of elements required by an array as follows:

```
var variable_name [SIZE] variable_type
```

For example, to declare a 10-element array called balance of type float32, use this statement:

```
var balance [10] float32
```

You can initialize array in Go either one by one or using a single statement as follows:

```
var balance = []float32{1000.0, 2.0, 3.4, 7.0, 50.0}
```

Slices

Golang Slice is an abstraction over Go Array. As Go Array allows you to define type of variables that can hold several data items of the same kind but it do not provide any inbuilt method to increase size of it dynamically or get a sub array of its own. Slices covers this limitation. It provides many utility functions required on Array and is widely used in Go programming.

To define a slice, you can declare it as an array without specifying size or use **make** function to create the one.

```
var numbers []int /* a slice of unspecified size */
/* numbers == []int{0,0,0,0,0}*/
numbers = make([]int,5,5) /* a slice of length 5 and capacity 5*/
```

Structure

structure in golang is a user defined data type available in Go programming, which also allows you to combine da

To define a structure, you must use **type** and **struct** statements. The struct statement defines a new data type, with more than one member for your program. type statement binds a name with the type which is struct in our case. The format of the struct statement is this

```
type struct_variable_type struct {
   member definition;
   member definition;
   ...
   member definition;
}
```

Once a structure type is defined, it can be used to declare variables of that type using following syntax.

```
variable_name := structure_variable_type {value1, value2...valuen}
```

Creating server in Golang

- net/http package
- HandleFunc
- Handler
- Chi Router

Demo

Q&A