

Golang Goroutines

Golang Goroutines

A goroutine is a lightweight thread managed by the Go runtime.

To start a goroutine, use keyword `go` before making the function call as shown below:

```
go somefunc(arg1, arg2)
```

When the above statement is executed, the statement itself gets executed in the current goroutine, but `somefunc(arg1, arg2)` gets executed in a new goroutine.

Example 1 – Goroutine

In the following program, we will implement a goroutine and call a function to print a message.

example.go

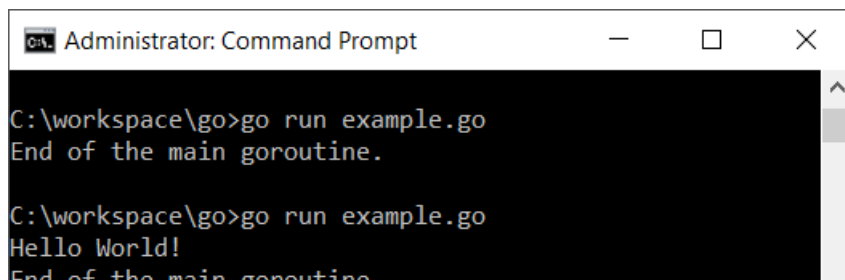
```
package main

import "fmt"

func printmessage(s string) {
    fmt.Println(s)
}

func main() {
    // a goroutine
    go printmessage("Hello World!")
    // another go routine
    go printmessage("Welcome to Golang Goroutines.")

    fmt.Println("End of the main goroutine.")
}
```



```
Administrator: Command Prompt
C:\workspace\go>go run example.go
End of the main goroutine.

C:\workspace\go>go run example.go
Hello World!
End of the main goroutine.
```

```
End of the main goroutine.  
C:\workspace\go>
```

Try running this program, using `go run` command in command prompt or terminal. You may or may not see the strings passed to `printmessage()` function in the goroutines printed to the console. It is because, when you run the program, the other goroutines as well run on the same address space and access the same shared memory and resources.

If the main goroutine or other goroutine is accessing console output to print the message, and if at the same time other goroutine checks for a console output device, it might find the console output to be in use. Hence, the string will not be displayed.

Conclusion

In this [Golang Tutorial](#), we learned about Golang Goroutines with the help of example programs.

Golang

- ◆ [Golang Tutorial](#)
- ◆ [Run Golang Program](#)
- ◆ [Golang If Else](#)
- ◆ [Golang Switch](#)
- ◆ [Golang For Loop](#)
- ◆ [Golang Comments](#)
- ◆ [Golang Functions](#)
- ◆ [Golang Array](#)
- ◆ [Golang Slice](#)
- ◆ [Golang Struct](#)
- ◆ [Golang Class](#)
- ◆ [Golang Range](#)
- ◆ [Golang Map](#)
- ⇒ [Golang Goroutine](#)
- ◆ [Golang Channel](#)

Golang String Operations

- ◆ [Golang String Length](#)

◆ [Golang String Concatenation](#)

◆ [Golang Split String](#)

◆ [Golang String - Get Index of Substr](#)

◆ [Golang String to Uppercase](#)

◆ [Golang String to Lowercase](#)

◆ [Golang Convert String to Integer](#)