

# Go Language Practice

---

## Start Go

### Hello World in Go

```
```go
package main

import "fmt"

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

---

### To run Go code in terminal:

```
```bash
@Satyams-MacBook-Air GO Language % go run main.go
Hello Go World!!
```
```

---

### Print Multiple Lines

```
```go
package main

import "fmt"

func main() {
    fmt.Println("Welcome to do list\n")
    fmt.Println("Watch go crash course")
    fmt.Println("Watch Nana's good boy ")
    fmt.Println("Rewards myself with a cheekcake")
}
```
```

---

## Data Types in Go

- Strings
- Booleans
- Integer
- Maps
- Arrays

---

### Example

```
```go
package main

import "fmt"

func main() {
    fmt.Println("###Welcome to do list!###")
    fmt.Println("1. Watch go crash course")
    fmt.Println("2. Watch Nana's good boy ")
    fmt.Println("3. Rewards myself with a cheekcake")
    fmt.Println(4)
    fmt.Println(5)
}
```
```

---

## ## Storing Values in Variables

```
```go
package main

import "fmt"

func main() {
    var taskone = "2. Watch Nana's good boy "

    fmt.Println("###Welcome to do list!###")
    fmt.Println("1. Watch go crash course")
    fmt.Println(taskone)
    fmt.Println("3. Rewards myself with a cheekcake")
    fmt.Println(4)
    fmt.Println(5)
}
```
```

---

## ## List Data Types in Go

1. Arrays
2. Slice

```
```go
package main

import "fmt"

func main() {
    // var taskone = "2. Watch Nana's good boy "
    // var tasktwo="1. Watch go crash course"
    // var maxItemsInGroup=20

    // below line is slice
    var taskitem = []string{"Watch Nana's good boy ", "Watch go crash course", "Learn Go", "Learn Python"}

    fmt.Println("###Welcome to do list!###")
    fmt.Println()
    fmt.Println(taskitem)
}
```
```

---

## ### Slice Example with Variables

```
```go
package main

import "fmt"

func main() {
    var taskone = "2. Watch Nana's good boy "
    var tasktwo = "1. Watch go crash course"
    var maxItemsInGroup = 20

    var taskitem = []string{taskone, tasktwo}

    fmt.Println("###Welcome to do list!###")
    fmt.Println()
    fmt.Println(taskitem)
    fmt.Println(maxItemsInGroup)
}
```
```

---

## ## Array vs Slice

```

```go
// this is slice where size is not defined
var taskitem = []string{taskone,tasktwo}

// this is array where size is defined
var taskitem = [2]string{taskone,tasktwo}
```

---

## Loops in Go

### Types:
1. Basic for loop
2. For loop with condition
3. For range loop

---

### Example (Range Loop)

```go
package main

import "fmt"

func main() {
    var taskone = "2. Watch Nana's good boy "
    var tasktwo = "1. Watch go crash course"
    var taskthree = "3. Watch Naruto"

    var taskitem = []string{taskone,tasktwo,taskthree}

    for _,task := range taskitem {
        fmt.Println(task)
    }
}
```

---

### Example with Index

```go
package main

import "fmt"

func main() {
    var taskone = " Watch Nana's good boy "
    var tasktwo = " Watch go crash course"
    var taskthree = " Watch Naruto"

    var taskitem = []string{taskone,tasktwo,taskthree}

    for index, task := range taskitem {
        fmt.Println(index, task)
    }
}
```

---

### Formatted Output

```go
package main

import "fmt"

func main() {
    var taskone = " Watch Nana's good boy "

```

```

    var tasktwo = " Watch go crash course"
    var taskthree = " Watch Naruto"

    var taskitem = []string{taskone,tasktwo,taskthree}

    for index, task := range taskitem {
        fmt.Printf("%d. %s\n", index+1, task)
    }
}
...

```

---

## ## Functions in Go

```

```go
package main

import "fmt"

var taskone = " Watch Nana's good boy "
var tasktwo = " Watch go crash course"
var taskthree = " Watch Naruto"

var taskItems = []string{taskone, tasktwo, taskthree}

func main() {
    fmt.Println("### Welcome to Task Manager ###")
    // printTasks()
}

func printTasks() {
    fmt.Println("Printing tasks")
    for index, task := range taskItems {
        fmt.Printf("%d: %s\n", index+1, task)
    }
}
...

```

---

## ## Variable Scope & Global Scope

```

```go
package main

import "fmt"

var taskone = " Watch Nana's good boy "
var tasktwo = " Watch go crash course"
var taskthree = " Watch Naruto"

var taskItems = []string{taskone, tasktwo, taskthree}

func main() {
    fmt.Println("### Welcome to Task Manager ###")
    printTasks()
}

func printTasks() {
    fmt.Println("Printing tasks")
    for index, task := range taskItems {
        fmt.Printf("%d: %s\n", index+1, task)
    }
}
...

```

---

## ## Passing Values Between Functions

```

```go

```

```

package main

import "fmt"

func main() {
    fmt.Println("### Welcome to Task Manager ###")
    var taskone = " Watch Nana's good boy "
    var tasktwo = " Watch go crash course"
    var taskthree = " Watch Naruto"

    var taskItems = []string{taskone, tasktwo, taskthree}
    printTasks(taskItems)
}

func printTasks(taskItems []string) {
    fmt.Println("Printing tasks")
    for index, task := range taskItems {
        fmt.Printf("%d: %s\n", index+1, task)
    }
}

```

---

## Working with List Items

```

```go
package main

import "fmt"

func main() {
    fmt.Println("### Welcome to Task Manager ###")
    var taskone = " Watch Nana's good boy "
    var tasktwo = " Watch go crash course"
    var taskthree = " Watch Naruto"

    var taskItems = []string{taskone, tasktwo, taskthree}
    printTasks(taskItems)
    fmt.Println()
    addTask(taskItems, " Let's Run")
}

func printTasks(taskItems []string) {
    fmt.Println("Printing tasks")
    for index, task := range taskItems {
        fmt.Printf("%d: %s\n", index+1, task)
    }
}

func addTask(taskItems []string, newTask string) {
    var updatedTaskItems = append(taskItems, newTask)
    printTasks(updatedTaskItems)
}

```

---

## Returning Values from Functions

```

```go
package main

import "fmt"

func main() {
    fmt.Println("### Welcome to Task Manager ###")
    var taskone = " Watch Nana's good boy "
    var tasktwo = " Watch go crash course"
    var taskthree = " Watch Naruto"

    var taskItems = []string{taskone, tasktwo, taskthree}

```

```

    printTasks(taskItems)
    fmt.Println()

    taskItems = addTask(taskItems, " Let's Run")
    fmt.Println()
    printTasks(taskItems)
}

func printTasks(taskItems []string) {
    fmt.Println("Printing tasks")
    for index, task := range taskItems {
        fmt.Printf("%d: %s\n", index+1, task)
    }
}

func addTask(taskItems []string, newTask string) []string {
    var updatedTaskItems = append(taskItems, newTask)
    return updatedTaskItems
}
...

---

## Web API in Go

### Example 1

```go
package main

import (
    "fmt"
    "net/http"
)

func main() {
    http.HandleFunc("/", helloUser)

    fmt.Println("Server starting at http://localhost:8080")
    http.ListenAndServe(":8080", nil)
}

func helloUser(writer http.ResponseWriter, r *http.Request) {
    greeting := "Hello User, Welcome to Task Manager"
    fmt.Fprintln(writer, greeting)
}
...

```

➡■ Open browser and type: `http://localhost:8080/`

```

---

### Example 2

```go
package main

import (
    "fmt"
    "net/http"
)

func main() {
    http.HandleFunc("/hello-go", helloUser)

    fmt.Println("Server starting at http://localhost:8080")
    http.ListenAndServe(":8080", nil)
}

func helloUser(writer http.ResponseWriter, r *http.Request) {
    greeting := "Hello User, Welcome to Task Manager"
    fmt.Fprintln(writer, greeting)
}

```

```

}
...

---

## Multiple Routes Example

```go
package main

import (
    "fmt"
    "net/http"
)

var shortgo = "Watch crash course"
var fullgolang = "Watch go lang full course"
var reward = "Reward yourself with a chocolate"

var taskItems = []string{shortgo, fullgolang, reward}

func main() {
    http.HandleFunc("/hello-go", helloUser)
    http.HandleFunc("/show-tasks", showTasks)

    fmt.Println("Server starting at http://localhost:8080")
    http.ListenAndServe(":8080", nil)
}

func showTasks(writer http.ResponseWriter, r *http.Request) {
    for _, task := range taskItems {
        fmt.Fprintln(writer, task)
    }
}

func helloUser(writer http.ResponseWriter, r *http.Request) {
    greeting := "Hello User, Welcome to Task Manager"
    fmt.Fprintln(writer, greeting)
}
...

---

### Output

- [http://localhost:8080/show-tasks](http://localhost:8080/show-tasks)
...
Watch crash course
Watch go lang full course
Reward yourself with a chocolate
...

- [http://localhost:8080/hello-go](http://localhost:8080/hello-go)
...
Hello User, Welcome to Task Manager
...

```