

# Go Command Line Argument

When we need to execute a program with some arguments, we generally use command line argument. The arguments passed from the console can be received by the Go program and it can be used as an input.

- ➔ The `os.Args` is used to get the arguments.
- ➔ The index 0 of `os.Args` contains the path of the program.
- ➔ The `os.Args[1:]` holds provided arguments.

Example 1:-

```
package main

import (
    "fmt"
    "os"
)

func main() {
    var s, arg string
    var i int

    for i = 1; i < len(os.Args); i++ {
        s += arg + os.Args[i] + " "
    }
    fmt.Println(s)
}
```

Command to run the program :-

- ➔ Open command prompt or Terminal
- ➔ Navigate to the folder where your program is saved
- ➔ Then build the program and give command line argument to it

Example :-

```
go build Example1.go
./Example1 Hello Gautam
```

**Note :- Example1 is the name of program**

**Output :-**

**Hello Gautam**

**Example 2 :-**

```
package main

import (
    "fmt"
    "os"
)

func main() {
    var argumentat1, argumentat2 string

    argumentwithpath := os.Args // return all
arguments including path
    argumentat1 = os.Args[1]    // return argumant
after path
    argumentat2 = os.Args[2]    // return specified
argument only

    fmt.Println(argumentwithpath)
    fmt.Println(argumentat1)
    fmt.Println(argumentat2)
}
```

**→ Build and give argument again in terminal or CMD**

```
PS C:\Users\Gautam Jha\Desktop\Git_Hub\Go_Language\16. Command_Line_args_in_Go> go build Example2.go
PS C:\Users\Gautam Jha\Desktop\Git_Hub\Go_Language\16. Command_Line_args_in_Go> ./Example2 I am Gautam
Jha Learning Go Language.
```

### **Output:-**

```
[C:\Users\Gautam Jha\Desktop\Git_Hub\Go_Language\16. Command_Line_args_in_Go\Example2.exe I am Gautam  
Jha Learning Go Language.]  
I  
am
```