

Go on HPC

What is Go?

Go is expressive, concise, clean, and efficient. Its concurrency mechanisms make it easy to write programs that get the most out of multicore and networked machines, while its novel type system enables flexible and modular program construction. Go compiles quickly to machine code yet has the convenience of garbage collection and the power of run-time reflection. It's a fast, statically typed, compiled language that feels like a dynamically typed, interpreted language. [\[source\]](#)

Links:

[Official Website](#)

[Documentation](#)

Versions Available:

The following versions are available on the cluster:

- go1.16.2

How to load Go?

To load Go, use the following commands:

```
#Load the Go module  
module load go/1.16.2
```

To verify if the module and dependencies are loaded correctly, use the following command.

```
#Show all the modules loaded  
module list
```

This should list all the Go and dependencies that are loaded. Since this is a standalone software, there should be not dependencies.

How to use Go?

To demonstrate the usage of Go, use the following code snippet

```
package main  
  
import "fmt"  
  
func sum(nums ...int) {  
    fmt.Print(nums, " ")  
    total := 0  
  
    for _, num := range nums {  
        total += num  
    }  
    fmt.Println(total)  
}  
  
func main() {  
  
    sum(1, 2)  
    sum(1, 2, 3)  
  
    nums := []int{1, 2, 3, 4}  
    sum(nums...)  
}  
  
#SOURCE
```

Make a go file in home directory and paste the following code in it,

```
# Make a go file
touch example.go
```

Run the go file by using the following command,

```
# Run with go
go run example.go
```

Output should be printed in stdout as,

```
[1 2] 3
[1 2 3] 6
[1 2 3 4] 10
```

Refer to the Go docs for more examples.

Where to find help?

If you are stuck on some part or need help at any point, please contact OIT at the following address.

<https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp>

