

de Porfideacises
acvited
gendito/ Penel))
ecesattlnajof
ystund.
acesstlgrgstconrserter
agelock
odalreterassconrserter
upcatonMattide
ogallerargerconrserter
eopsettins
odallertsterconrserter
ofiun: fastzing
ocesstlgrperconrserter
reldings.
ocesstlgrApoin
chonaplag to
l forpani
acesstlgrconrserter
cr/onlt
ocester
olllin
ogeseti
acesellgr
each
goecatigyres
e:
acesellgracncfrtea
dure
acesstlgrsterconrserter
essude..
ocagalloystapperartter
engs.
ocesstlgrperconrsertton
oytaating:
togañl lgo:
dled.co repany

GO

Golang (Go) Programming Language

SULTAN - 56189 - BS/DS-5_1



Introduction to Go

Developed by Google

Created in 2007, released 2009 by
Griesemer, Pike, Thompson.

Efficiency & Simplicity

Combines C's efficiency with
Python's simplicity.

Scalability & Performance

Statically typed, compiled for
modern cloud, server, distributed
apps.

Uses of Golang



Backend Web Development

Building robust and scalable server-side applications.



Cloud Infrastructure

Powering cloud services and platforms.



DevOps Tools

Creating efficient tools for development and operations.



Network Programming

Ideal for high-performance network applications.

Go's concurrency with goroutines and channels handles multiple tasks efficiently.

Go in Data Science

While not as dominant as Python, Go is gaining traction for:

- High-performance data processing
- Real-time analytics
- Building scalable data pipelines

Its speed and concurrency are suitable for big data and streaming microservices.



A stylized illustration of a server rack with multiple blue doors, situated within a landscape of soft, white, puffy clouds. Thin, grey lines representing cables or data paths connect the server rack to various points in the cloud environment, including a small cloud icon in the upper left and a circular node with radiating lines in the lower right.

Real-Life Example: Docker & Kubernetes



Docker

Containerization platform built with Go.



Kubernetes

Orchestration system also written in Go.

These Go-powered technologies revolutionized cloud computing and container orchestration, showcasing Go's simplicity, speed, and scalability for high-performance backend systems.

Comparison: Go vs Python vs C++

Typing	Static	Dynamic	Static
Compilation	Compiled	Interpreted	Compiled
Speed	Fast	Moderate	Very Fast
Ease of Use	Easy	Very Easy	Moderate
Concurrency	Built-in (goroutines)	Threading (limited)	Threading/Manual
Use Case	Cloud, Backend, Networking	AI, Data Science, Automation	System Software, Games

Sample Golang Syntax

```
package main
import "fmt"
func main() {
    fmt.Println("Hello World")}
}
```

This simple "Hello World" program demonstrates Go's clean and concise syntax.



```
1 package main
2 import "fmt"
3 func main() {
4     fmt.Println("Hello World")
5 }
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

that putated Weepe:
Correcteticats: wart
qually:

Hello World

purcaon eillle, -enegies ()
{ goo
{ accourallert-clapentles:
/ accoMeldrar-porhacice:
/ coo
/ accouralltart layp
/ accoMaltanendngsle,oho-coclonmet: any /abueleniond tape,storl
/ randllar:
{ accoMhelore -correctnent)
/ accolecal/ogae-Airt paretute cuso.lo:

/ accourol/ISky:
. auact/olcadil, omuntion:



Conclusion

Powerful & Efficient

Bridges simplicity and performance.

Modern Design

Concurrency model, cross-platform capabilities.

Ideal for Scalable Apps

Cloud, networking, backend systems.

Emerging in Data Science

Immense potential in data engineering.

GITHUB LINK: <https://github.com/sultanali543/INTRO-TO-GOLANG-GO->

THANK
YOU
FOR YOUR
TIME

