

# Go Training

## Session 1

# Go Introduction

---

- Go/Golang : Free and Open Source
- Created at and maintained by Google
- Launched in the year 2009
- System programming language

# Go Programming Syntax

- Semicolon are implicit at the end of a line
- Commenting styles
- Combined declaration /initialization operator

`x := 5`

A screenshot of a code editor window titled 'hello.go'. The editor shows a Go program with line numbers 1 through 10 on the left. The code is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     // prints Hello World
7     fmt.Println("Hello World")
8 }
9
10
```

The code is color-coded: 'package' is orange, 'main' is white, 'import' is orange, 'fmt' is green, 'func' is orange, 'main()' is white, '{' is white, '//' is white, 'prints' is white, 'Hello' is white, 'World' is green, and 'Println()' is white. The closing brace '}' is white.

# What GO offers and what it does not

---

- Statically typed, compiled language : like C++/Java
- First Class Function; Package system
- Allows type Inference
- Supports inbuilt concurrency
- Garbage collection : automatic memory management
- Strings are Immutable and are UTF-8 encoded
- Return multiple value from functions
- Great Standard library
- Defer statement

# What GO offers and what it does not

- NOT allows implicit numeric conversion
- Does not have CLASSES
- Does not have EXCEPTIONS
- NO function overloading
- GO doesn't strictly support OOP
  - However it is a lightweight object oriented language

← → ↻ 🔒 [golang.org/doc/faq](https://golang.org/doc/faq)

## Is Go an object-oriented language? ¶

Yes and no. Although Go has types and methods and allows an object-oriented style of programming, there is no type hierarchy. The concept of “interface” in Go provides a different approach that we believe is easy to use and in some ways more general. There are also ways to embed types in other types to provide something analogous—but not identical—to subclassing. Moreover, methods in Go are more general than in C++ or Java: they can be defined for any sort of data, even built-in types such as plain, “unboxed” integers. They are not restricted to structs (classes).

Also, the lack of a type hierarchy makes “objects” in Go feel much more lightweight than in languages such as C++ or Java.

# GO vs Other languages

---

- Simpler syntax compared to other compiled languages
- C/C++
  - Good at Speed, scale and reliability
  - NOT that good at simplicity
- Java
  - Good at scale and reliability
  - Moderately simple
  - Moderately efficient/fast
- Python
  - Scalable, very simple
  - NOT so efficient and reliable
- Golang
  - Gives you a feel of writing a scripting language
    - Without compromising the essence of low-level programming language

# Hello World Program

```
hello.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     // prints Hello World
7     fmt.Println("Hello World")
8 }
9
10
```

# Compiling and running

---

- `go run hello.go`
- `go build hello.go`
  - `./hello`
- `Go build -o newName hello.go`
  - `./newName`



# Install GO

- [Golang.org/doc/install](https://golang.org/doc/install)

← → ↻ 🔒 [golang.org/doc/install](https://golang.org/doc/install)

## 2. Go install.

Select the tab for your computer's operating system below, then follow its installation instructions.

Linux

Mac

Windows

1. Extract the archive you downloaded into `/usr/local`, creating a Go tree in `/usr/local/go`.

**Important:** This step will remove a previous installation at `/usr/local/go`, if any, prior to extracting. Please back up any data before proceeding.

For example, run the following as root or through `sudo`:

```
rm -rf /usr/local/go && tar -C /usr/local -xzf go1.16.3.linux-amd64.tar.gz
```

2. Add `/usr/local/go/bin` to the `PATH` environment variable.

You can do this by adding the following line to your `$HOME/.profile` or `/etc/profile` (for a system-wide installation):

```
export PATH=$PATH:/usr/local/go/bin
```

**Note:** Changes made to a profile file may not apply until the next time you log into your computer. To apply the changes immediately, just run the shell commands directly or execute them from the profile using a command such as `source $HOME/.profile`.

3. Verify that you've installed Go by opening a command prompt and typing the following command:

---

# Thank You