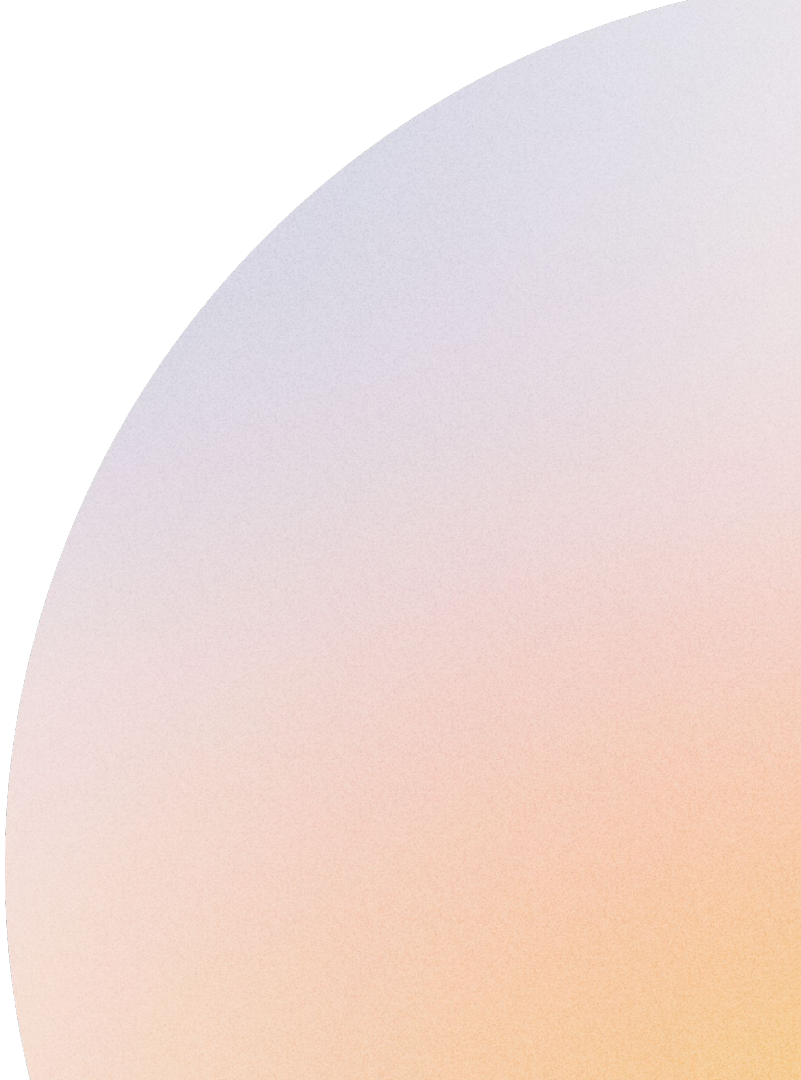
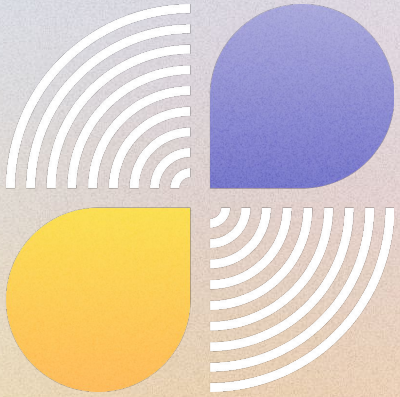
# Introduction to the Go language

## Ambush Journey Program

1. The Go Language
2. Why Go?
3. Installation & Setup
4. Practice





# The Go Language

# What is Go?

Go is an open source, multi-paradigm language that is sometimes described as an "improved C", with features that make it very powerful.

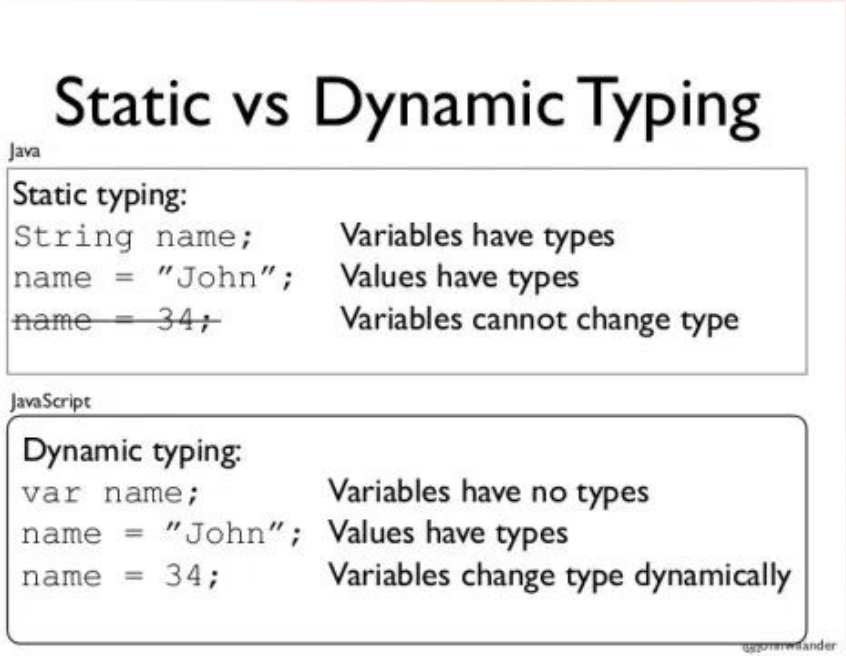
# What is Go?

A few of Go's features:

- It is **statically typed**
- It is high-level compiled
- It has **memory safety**
- It has **garbage collection**
- It has **structural typing**
- It has support to **concurrency**

# Features from Go

- It is **statically typed**
  - Just like C, Java or Swift
  - But **not** like Python or Javascript
- Static typing helps preventing runtime errors and provides better performance, among other benefits



# Features from Go

- It has **memory safety**
  - Go produces runtime errors when access to invalid / non allocated buffers of memory is made
    - For instance, when trying to access the data from a pointer that hasn't been allocated
  - This improves security as it makes the language fail-safer and prevents access to sensitive data
- Similar to Java, but not similar to C or C++

# Features from Go

- It has **garbage collection**
  - Programmers don't need to free memory space used by applications when those are not needed anymore; the garbage collector does this automatically.
  - Improves code writing and reduces boilerplate code
- Other languages with garbage collection: Python, Java
- Languages without garbage collection: C and C++

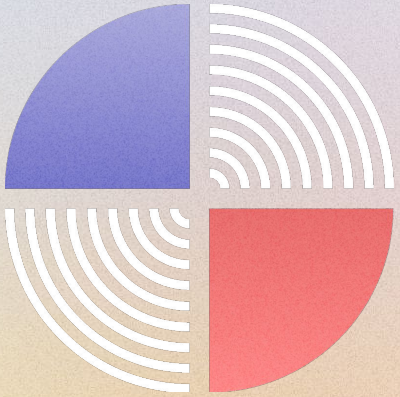
# Features from Go

- It has **structural typing**
  - An interface can be implemented by two types, and any objects that obey that interface will also obey those types

```
interface IPerson {
  age: number
}

function stringifyAge(value: IPerson): string {
  return `${value.age} years old`;
}

You, seconds ago | 1 author (You)
interface ICar {
  age: number,
  model: string,
}

const car: ICar = {age: 10, model: 'Model'};

// it works
stringifyAge(car);
```

# Features from Go

- It has support to **concurrency**
  - Two different blocks of code may be executed simultaneously
  - Probably the most powerful feature of Go, as it greatly improves performance
  - Concurrency is made through a feature called **goroutines**.
    - We will look into this in the future

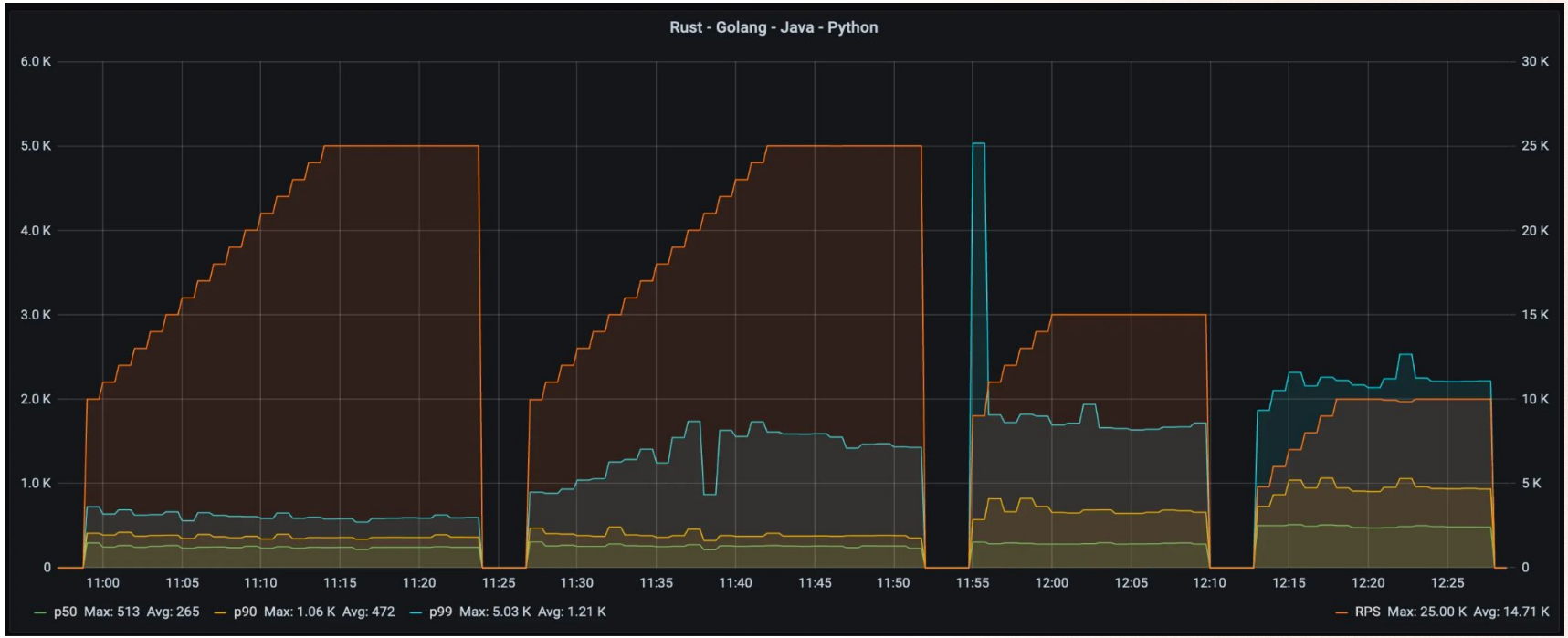


# Why Go?

# Why Go?

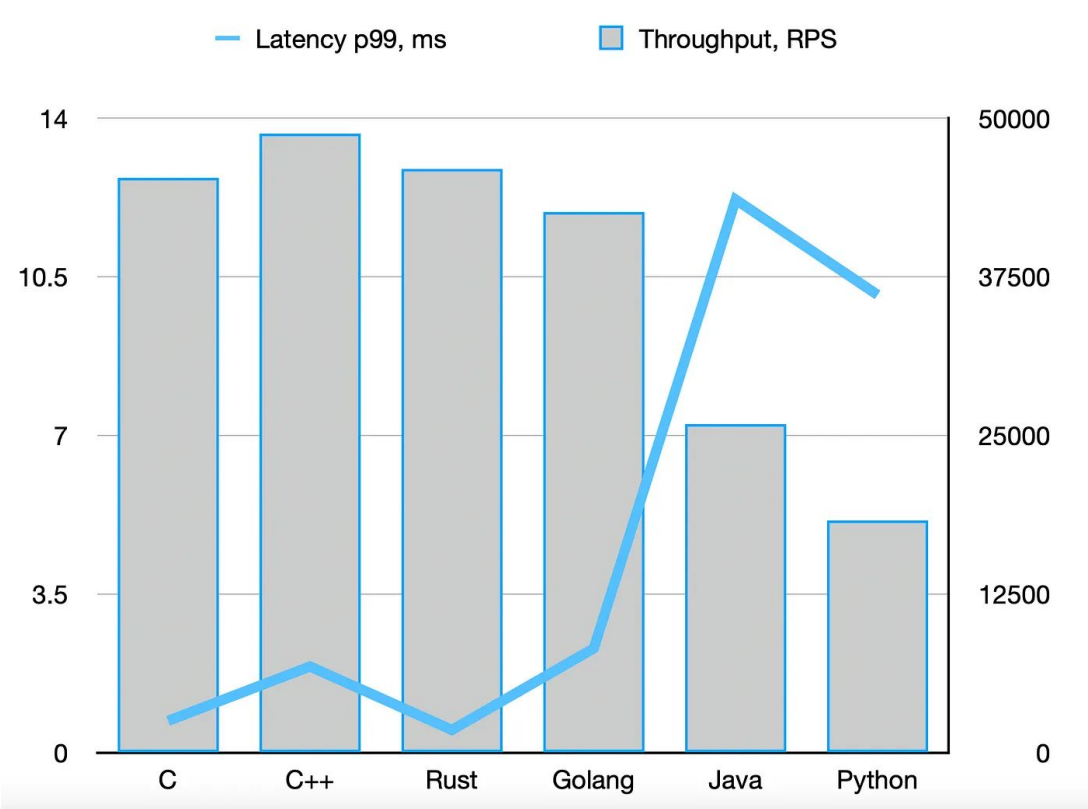
- All the features previously listed make Go one of the most **secure** and **fastest** programming languages, which is very suitable for backend servers.
- It's fairly simple and easy to learn, including the easiness of building concurrent applications.
- Handles network I/O efficiently.
- It's very scalable and easy to deploy.

# Why Go?




Source: Medium.com (Benchmarking low-level I/O: C, C++, Rust, Golang, Java, Python)

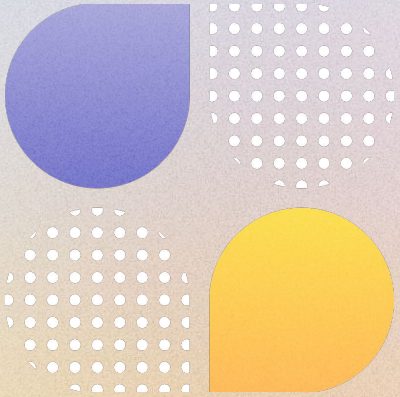
# Why Go?




Source: Medium.com (Benchmarking low-level I/O: C, C++, Rust, Golang, Java, Python)

# Why Go?

- Further reading: the official website includes a list of use cases and scenarios where companies use/have used Go



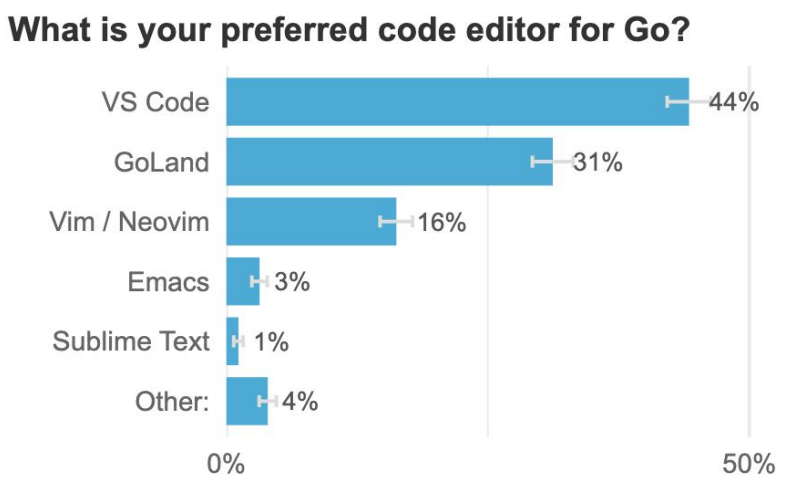
# Installation & Setup

# Installation

- First, download Go at its official website
  - There are specific steps (listen in there) depending on your operating system
  - Once you have installed the package, you should be able to verify that it is working by opening a terminal or command prompt and typing in the following command:

# Setup

- After installing Go in your machine, you should pick an editor to write Go code.




Source: [Go Developer Survey 2023]

# Setup

- After installing Go in your machine, you should pick an editor to write Go code.
  - There are multiple programs available, but the most used ones are VS Code, GoLand and VIM.
  - Since VIM is a unix-terminal editor and not very easy to use, we will briefly focus on the first two only.

# Setup - VS Code

Visual Studio Code (VS Code) is a programming IDE created by Microsoft. A few of its advantages are:

- It's free to use
- It supports not only Go, but almost any programming language (which is useful if you are coding a FS application for instance)
- It has a big community and allows users to create custom libraries to be used with it
- It's good for starters

# Setup - GoLand

GoLand is a programming IDE created by JetBrains. A few of its advantages are:

- Since it's made specifically for Go, it provides a lot of useful features such as code auto-completion, package recognition, refactoring, among others
- Very efficient for production development
- Even though it's not free, users may be able to get it with any student license (any valid `.edu` e-mail for instance)

# Setup

- VS Code and GoLand can be downloaded and installed through their official websites (VS Code and GoLand)
- Besides these two, any lightweight editor may work for simple tasks (such as Sublime Text or Notepad++)

# Practice

# Practice

- After you have installed Go in your machine, along with an editor of your choice (we recommend VS Code for these first practice classes), let's write a simple hello world application
- Any files with the `.go` extension can be ran by the Go compiler
- In order to run a go file, just type in `go run` and the file name

```
→  playground git:(feature/create-first-lesson) ✗ go run hello_world.go
Welcome to Go programming!
Hello, World!
```

# Practice

A few notes:

- Any executable Go program should have the base structure shown below:

# Practice

A few notes:

- The `fmt` package can be used to print text in the console
- The `package main` directive allows the code to be executed, calling the `main` function in the same file
- Package documentation can be found in the go package website
  - For instance, this link contains the documentation on the `fmt` package
  - Methods, variables, functions, and other info can be found there

# Introduction to the Go language

## Ambush Journey Program