**Nhóm 10:**

**Quách Xuân Hoàng – 3120410184**

**Bùi Minh Trí – 3120410547**

**Lưu Gia Tính – 3120410536**

**Huỳnh Xiêm – 3120410634**

**Tìm hiều về ngôn ngữ lập trình Go**

1. **Lịch sử**

Ngôn ngữ lập trình Go hay còn được gọi là Golang là một ngôn ngữ lập trình được phát triển bởi Google, bởi nhóm kỹ sư bao gồm Robert Griesemer, Rob Pike và Ken Thompson vào ngày 21 tháng 9 năm 2007.

Phiên bản đầu tiên được phát hành vào ngày 28 tháng 3 năm 2012, được gọi là Go 1. Đây được xem là bản phát hành ổn định đầu tiên của ngôn ngữ này.

Sau đó liên tục các bản cập nhật và nâng các trải dài cho đến phiên bản Go 1.21.1 được sử dụng hiện nay.

Go được đánh giá cao về tính ổn định, hiệu suất và khả năng đồng thời xử lý (concurrency), điều này đã giúp nó trở thành một trong những lựa chọn phổ biến cho các hệ thống có yêu cầu cao về độ tin cậy và hiệu suất.

1. **Mục đích**

Ngôn ngữ lập trình Go (Golang) được phát triển với mục đích cung cấp một công cụ mạnh mẽ và hiệu quả cho việc phát triển phần mềm. Dưới đây là một số mục đích chính của Go:

* **Tối ưu hóa hiệu suất**: Một trong những mục tiêu quan trọng của Go là đảm bảo hiệu suất cao. Go được thiết kế để thực thi nhanh và tiêu tốn ít tài nguyên hệ thống, điều này làm cho nó trở thành một lựa chọn tốt cho các ứng dụng yêu cầu xử lý nhanh như máy chủ web, dịch vụ mạng và hệ thống phân tán.
* **Đơn giản hóa:** Go chú trọng vào sự đơn giản và dễ đọc. Ngôn ngữ này giảm đi một số tính năng phức tạp và cú pháp khó hiểu mà các ngôn ngữ khác có thể có. Điều này giúp tạo ra mã nguồn dễ bảo trì và độ tin cậy cao.
* **Hỗ trợ đồng thời xử lý (Concurrency):** Go cung cấp hỗ trợ đặc biệt cho đồng thời xử lý (concurrency) thông qua các goroutines và channels. Điều này giúp lập trình viên xây dựng các ứng dụng có khả năng xử lý nhiều công việc cùng một lúc một cách dễ dàng và an toàn.
* **Hỗ trợ xây dựng hệ thống lớn**: Go được thiết kế để phục vụ cho việc xây dựng các hệ thống lớn và phức tạp. Sự hỗ trợ cho các gói (packages) và các quy tắc tổ chức mã nguồn giúp việc phát triển các dự án lớn trở nên dễ dàng hơn.
* **Tích hợp dễ dàng**: Go hỗ trợ tích hợp dễ dàng với các thư viện và dự án khác. Với một hệ sinh thái phong phú của các thư viện và công cụ bên ngoài, Go giúp lập trình viên xây dựng ứng dụng nhanh chóng và hiệu quả.
* **Đảm bảo độ tin cậy và an toàn**: Go có một hệ thống kiểm tra kiểu (type system) mạnh mẽ và cung cấp các công cụ kiểm tra mã nguồn tự động. Điều này giúp đảm bảo tính đúng đắn và an toàn trong quá trình thực thi chương trình.

Tóm lại, mục đích chính của Go là cung cấp cho lập trình viên một ngôn ngữ đơn giản, hiệu quả và đáng tin cậy để xây dựng các ứng dụng và hệ thống phức tạp, đặc biệt là những ứng dụng đòi hỏi hiệu suất cao và khả năng đồng thời xử lý.

1. **Tính năng**

**Các kiểu dữ liệu cơ bản:**

* **Số nguyên**: Go hỗ trợ kiểu dữ liệu nguyên như int, int8, int16, int32, int64 và kiểu số nguyên không dấu như uint, uint8, uint16, uint32, uint64.
* **Số thực**: Go hỗ trợ kiểu dữ liệu số thực như float32 và float64.
* **Kiểu boolean**: bool.
* **Kiểu ký tự: rune** (dùng để đại diện cho một ký tự Unicode).
* **Kiểu chuỗi**: string.
* **Biến và hằng số**: Go cho phép khai báo biến và hằng số. Biến có thể được khai báo với từ khóa var, và hằng số được khai báo bằng từ khóa const.

**Câu lệnh điều kiện:**

**if**: Sử dụng để kiểm tra một điều kiện và thực hiện một khối mã nếu điều kiện đúng.

**switch:** Sử dụng cho kiểm tra nhiều trường hợp khác nhau.

**select:** Sử dụng cho kiểm tra và chọn một trong các lựa chọn trong trường hợp có nhiều trường hợp đang chờ xử lý.

**Vòng lặp:**

**for**: Sử dụng để thực hiện vòng lặp.

**range**: Sử dụng để lặp qua các phần tử của một mảng, slice, map hoặc kênh (channel).

Các cấu trúc dữ liệu:

**Mảng và Slice**: Cho phép lưu trữ nhiều giá trị có cùng kiểu dữ liệu. Slice là một dạng linh hoạt của mảng.

**Map**: Dùng để lưu trữ các cặp key-value không theo thứ tự.

**Struct**: Cho phép tạo ra các kiểu dữ liệu tự định nghĩa bằng cách kết hợp nhiều kiểu dữ liệu khác.

**Hàm:**

Go hỗ trợ việc định nghĩa và sử dụng hàm.

Hàm có thể có nhiều tham số và giá trị trả về.

**Goroutines và Channels**:

**Goroutines** là một cơ chế cho đồng thời xử lý (**concurrency**) trong Go.

**Channels** là cách truyền thông tin và dữ liệu giữa các goroutines, giúp đảm bảo đồng bộ và an toàn trong xử lý đồng thời.

**Gói (Package):**

Go sử dụng khái niệm gói để tổ chức và quản lý mã nguồn.

Gói giúp tạo ra các thư viện và modules có thể tái sử dụng trong các dự án khác nhau.

Bảo mật và kiểm tra kiểu (Type Safety):

Go có một hệ thống kiểm tra kiểu mạnh mẽ để đảm bảo tính đúng đắn của mã nguồn.

Go cũng cung cấp các công cụ bảo mật như xử lý chuỗi và số nguyên một cách an toàn để tránh các lỗ hổng bảo mật.

1. **Tính năng**

Dưới đây là một số tính năng quan trọng của Go:

**Hỗ trợ đồng thời xử lý (Concurrency):**

Go cung cấp goroutines và channels để hỗ trợ đồng thời xử lý một cách dễ dàng. Goroutines là các tiểu trình nhẹ, cho phép bạn thực hiện hàng trăm hoặc hàng nghìn goroutines trong một chương trình mà không cần nhiều tài nguyên.

Channels là cách truyền thông tin giữa các goroutines, giúp đảm bảo đồng bộ và an toàn trong xử lý đồng thời.

**Kiểm tra kiểu (Type Safety):**

Go có một hệ thống kiểm tra kiểu mạnh mẽ. Điều này giúp tránh được nhiều lỗi thường gặp như chia cho 0, truy cập bộ nhớ sai, hoặc gán giá trị kiểu không phù hợp.

Kiểm tra kiểu tĩnh giúp tăng tính đáng tin cậy và tránh được nhiều lỗi runtime.

**Hệ thống quản lý thư viện (Package System):**

Go có hệ thống quản lý thư viện mạnh mẽ, dựa trên gói (package) và thư mục. Thư viện và các gói có thể dễ dàng sử dụng và chia sẻ trong các dự án khác nhau.

Go cung cấp một cơ chế tự động để tải và quản lý các phụ thuộc (dependencies) của dự án.

**Hỗ trợ đồ họa (Garbage Collection):**

Go sử dụng bộ thu gom rác (garbage collector) để quản lý bộ nhớ tự động. Điều này giúp ngăn chặn rò rỉ bộ nhớ và giảm khó khăn trong việc quản lý bộ nhớ thủ công.

**Hỗ trợ đa nền tảng (Cross-Platform):**

Go được thiết kế để hỗ trợ đa nền tảng. Bạn có thể viết mã Go trên một hệ thống và biên dịch nó để chạy trên nhiều nền tảng khác nhau mà không cần chỉnh sửa mã nguồn.

**Không yêu cầu dấu chấm phẩy (Semicolon):**

Go tự động xác định hết dòng (end-of-line) là nơi kết thúc một câu lệnh, không yêu cầu sử dụng dấu chấm phảy ở cuối mỗi câu lệnh. Điều này làm cho mã Go sáng sủa và dễ đọc hơn.

**Không hỗ trợ kế thừa (Inheritance):**

Go không hỗ trợ kế thừa trực tiếp như một số ngôn ngữ khác. Thay vào đó, nó sử dụng composition (tạo thành) thông qua các struct để xây dựng các kiểu dữ liệu phức tạp.

**Mã nguồn dễ đọc và đơn giản:**

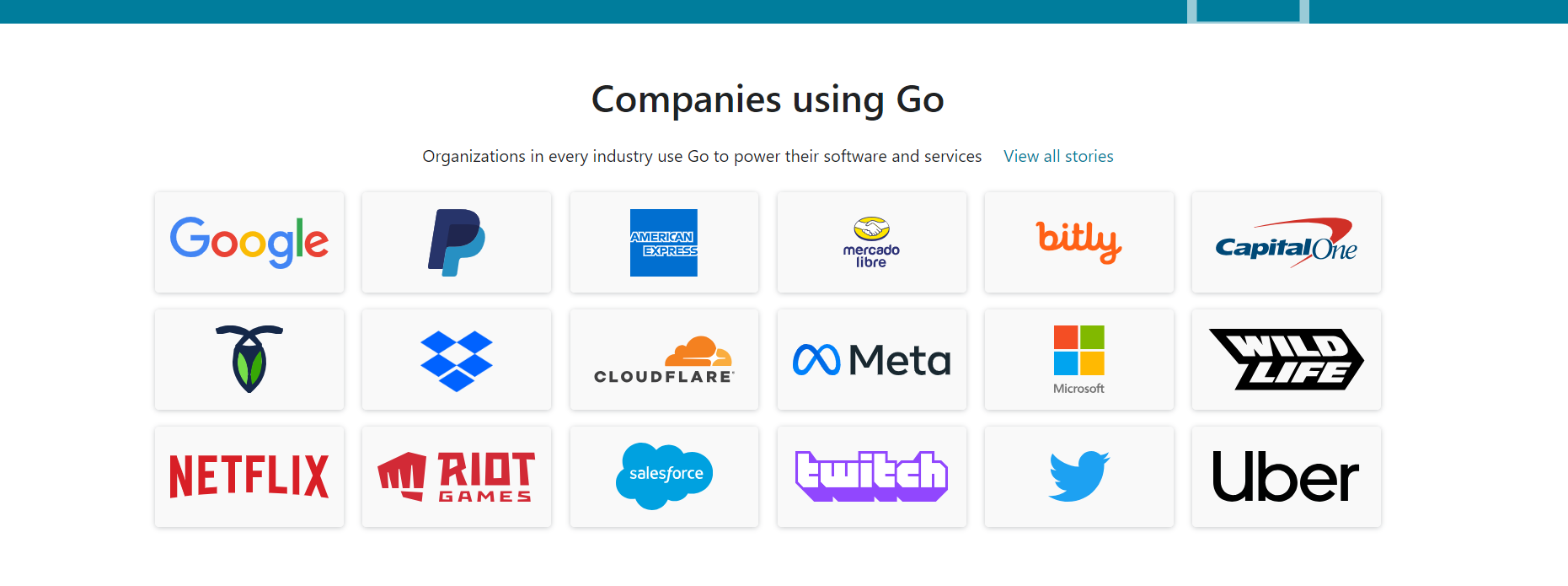
Go thiết kế để đảm bảo mã nguồn dễ đọc, với quy ước chặt chẽ về định dạng và cú pháp. Điều này giúp tăng tính bảo trì và độ tin cậy của mã nguồn.

**Hệ sinh thái mạnh mẽ:**

Go có một hệ sinh thái phong phú với nhiều thư viện và framework mạnh mẽ, bao gồm mạng, web, xử lý dữ liệu, và nhiều lĩnh vực khác.

1. **Ứng dụng**

Các ứng dụng đang sử dụng Go:



1. **Ưu và nhược điểm**

**Ưu điểm của Go:**

* **Hiệu suất cao**: Go được thiết kế để có hiệu suất cao. Mã Go thường thực thi nhanh hơn so với một số ngôn ngữ lập trình khác, đặc biệt là trong các ứng dụng có yêu cầu đòi hỏi xử lý nhanh như máy chủ web và dịch vụ mạng.
* **Hỗ trợ đồng thời xử lý mạnh mẽ**: Goroutines và channels làm cho việc xử lý đồng thời trở nên dễ dàng và hiệu quả, giúp tạo ra các ứng dụng có khả năng xử lý nhiều công việc cùng một lúc.
* **Kiểm tra kiểu mạnh mẽ**: Go có hệ thống kiểm tra kiểu mạnh mẽ giúp phát hiện lỗi kiểu tại thời điểm biên dịch, giúp tránh được nhiều lỗi runtime.
* **Mã nguồn dễ đọc và bảo trì**: Go thiết kế để đảm bảo mã nguồn dễ đọc và bảo trì, với cú pháp đơn giản và quy ước rõ ràng về định dạng.
* **Hệ sinh thái mạnh mẽ**: Go có một hệ sinh thái phong phú với nhiều thư viện và framework hữu ích cho nhiều loại ứng dụng.
* **Quản lý bộ nhớ tự động**: Go sử dụng thu gom rác (garbage collector) để quản lý bộ nhớ tự động, giúp tránh rò rỉ bộ nhớ và giảm khó khăn trong việc quản lý bộ nhớ thủ công.
* **Hỗ trợ đa nền tảng**: Go hỗ trợ đa nền tảng, cho phép bạn viết mã trên một hệ thống và biên dịch để chạy trên nhiều nền tảng khác nhau.

**Nhược điểm của Go:**

* **Không hỗ trợ kế thừa (Inheritance):** Go không hỗ trợ kế thừa trực tiếp, điều này có thể làm cho việc tái sử dụng mã nguồn trong một số trường hợp trở nên phức tạp hơn.
* **Thư viện hạn chế cho một số lĩnh vực cụ thể:** Mặc dù Go có một hệ sinh thái mạnh mẽ, nhưng có một số lĩnh vực cụ thể như lập trình đồ họa và thao tác với cơ sở dữ liệu mà thư viện và hỗ trợ không phong phú bằng một số ngôn ngữ khác.
* **Phát triển ứng dụng di động hạn chế:** Go không phải là một lựa chọn phổ biến cho việc phát triển ứng dụng di động, đặc biệt là trên nền tảng iOS.
* **Còn đang trong quá trình phát triển:** Mặc dù Go đã ra mắt từ nhiều năm, nhưng nó vẫn đang trong quá trình phát triển và có thể có thay đổi trong tương lai.

1. **Các nguồn tài liệu**

**Trang chính thức của Go:**

* Trang web chính thức của Go là nguồn tài liệu cơ bản và chính thống về ngôn ngữ này.
* Truy cập: <https://golang.org/>

**Go Playground:**

* Go Playground là một môi trường trực tuyến để thử nghiệm mã Go mà không cần cài đặt bất kỳ công cụ nào.
* Truy cập: <https://play.golang.org/>

**Tour of Go:**

* Tour of Go là một khóa học trực tuyến miễn phí cung cấp một hướng dẫn nhanh và thú vị để tìm hiểu về ngôn ngữ Go.
* Truy cập: <https://tour.golang.org/welcome/1>

**Go by Example:**

* Go by Example cung cấp ví dụ thực tế về cách sử dụng Go để giúp bạn hiểu rõ cú pháp và tính năng của ngôn ngữ.
* Truy cập: <https://gobyexample.com/>

**Golang.org Wiki:**

* Trang wiki trên trang web chính thức của Go chứa các tài liệu mở rộng về Go, bao gồm hướng dẫn, lời khuyên và tài liệu về các công cụ và thư viện.
* Truy cập: <https://github.com/golang/go/wiki>

**GoDoc:**

* GoDoc là một trang web cung cấp tài liệu tự động cho các thư viện Go. Bạn có thể tìm kiếm và xem tài liệu cho các gói cụ thể.
* Truy cập: <https://pkg.go.dev/>

**Các cuốn sách về Go:**

* Có nhiều cuốn sách về Go, từ sách hướng dẫn cơ bản đến sách về thiết kế ứng dụng và tối ưu hóa. Dưới đây là một số cuốn sách phổ biến:
* "The Go Programming Language" của Alan A. A. Donovan và Brian W. Kernighan.
* "Go in Action" của William Kennedy, Brian Ketelsen, và Erik St. Martin.
* "Programming in Go" của Mark Summerfield.

**Khóa học trực tuyến và video học:**

* Nhiều trang web và nền tảng giảng dạy trực tuyến cung cấp khóa học và video học về Go như Coursera, Udemy, edX, và Pluralsight.

**Cộng đồng Go:**

* Cộng đồng Go rất mạnh mẽ và hữu ích. Bạn có thể tham gia vào các diễn đàn, nhóm thảo luận trực tuyến, và thậm chí làm việc trên các dự án mã nguồn mở liên quan đến Go để học hỏi và gặp gỡ những người có kinh nghiệm.

1. **Các khóa học:**

Nhiều trang web và nền tảng giảng dạy trực tuyến cung cấp khóa học và video học về Go như Coursera, Udemy, edX, và Pluralsight.

Ngoài ra có thể tiếp cận các nguồn không mất phí trên youtube với từ khóa “học Golang”, “Golang course”, “Tour of go”, “Go tutorial”, …

1. **Tuyển dụng và mức lương**

Ta có thể tìm được khá nhiều cơ hội việc làm trên các trang tuyển dụng nổi tiếng như ITViec, TopDev, … Với các mức lương hấp dẫn dựa theo kinh nghiệm và quy mô dự án