

技術講習会

第1回 ゴーファーくんと学ぶGo言語の世界

2022年4月

XXX

XXX

この技術講習会(先端技術勉強会)について

- テーマはフリー、参加者から聞いたりして決めていきます。次回はAWS関係の予定です。
- 頻度は月1回?程度、基本1H。時間も未定ですが夕方がよさそうです。
- とりあえず始めてみて様子を見ながら続けます。
- 今回はスライドで資料を作りましたが、Web上のリソースなどの参照がメインになる回もあるかもしれません。
- 説明中も適宜間を挟むので、リアクションはその都度音声でもチャットでもOKです!

ゆるくやっていくので、気軽にいきましょう!



第1回 ゴーファーちゃんと学ぶ GO言語の世界

目的: Go言語の概要、
使いどころ、立ち位置など
基本情報を知ってもらう

ゴーファーちゃんです



ゴーフアーくんと学ぶ 言語の世界

1. Go言語とは
2. シェア的な話
3. Go言語の特徴
4. Go言語の特徴からくる→嬉しいところ
5. Go言語の使いどころと思われるシーン
6. 主要言語内での立ち位置①
7. 主要言語内での立ち位置②
8. 主要言語内での立ち位置③
9. Go言語の言語仕様上のあれこれ
10. Go言語の開発環境
11. まとめ

1. Go言語とは

本物の
gopher→
(ホリネズミ)



- ◆ Google発の21世紀のモダンな言語 <https://go.dev/>
 - ◆ 2009年登場。2012年v1.0、現在はv1.18
 - ◆ Google社内の標準言語7つ: C/C++、Java、JavaScript、Python、**Go**、TypeScript
 - ◆ “Better C”
 - ◆ 「Pythonの開発速度とCの実行速度」を目指した
 - ◆ “golang” “go-lang” 「ゴーラング」「ゴーラン」とも呼ばれる
 - ◆ 欠点: 単語“go”が一般的過ぎて**ググラビリティ(笑)**が低い!

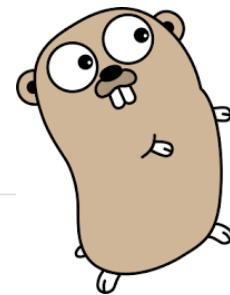


◆ 採用事例

- ◆ 公式より: Google、Dropbox、Meta(Facebook)、Microsoft、Netflix、Twitterなど
- ◆ 著名OSS: Docker、Kubernetes
- ◆ 日本: メルカリ/メルペイ、Gunosy、DeNA、サイバーエージェント、クックパッド、ぐるなび、Freee、はてな、クラスメソッド などなど
- ◆ インターネット系先進企業を中心に採用事例多し。
メルカリ社が積極フォローしていることで知られる。
- ◆ 雑誌でもよく取り上げられる。右はSoftware Design誌 2021/1号



2. Go言語のシェア的な話



◆シェア的なもの

- ◆Stackoverflowの2020年Most Lovedな言語5位、Wantedな言語3位
→これから学ぼうと注目されていることを表している。
- ◆求人検索エンジン「スタンプバイ」プログラミング言語別年収ランキング2018年の1位。→あてにならない？
- ◆Stackoverflowの同ランキング2020年でGlobal3位、USA内2位。
- ◆ネットの検索ランキングを元にしたTIOBE Index 2022年4月の言語ランキングで13位。Hall of Fame(殿堂入り)が2009年と2016年。
→検索が元なので若干信用度低し...シェアは低いが特定領域で注目されている。
- ◆ベンチャー、自社サービス系のエンジニアの求人広告でもよく見かける。デキるエンジニアが習得するイメージあり。2言語め以降で習得する人多し。
- ◆技術同人誌でも題材で時々見かける。
- ◆日本語で読める商業本では15冊ほど。



<https://insights.stackoverflow.com/survey/2020/#most-loved-dreaded-and-wanted>

<https://insights.stackoverflow.com/survey/2020/#technology-what-languages-are-associated-with-the-highest-salaries-worldwide-global>

6 <https://www.tiobe.com/tiobe-index/>

3. Go言語の特徴



- ◆①OSごとのネイティブコードへコンパイルされる
 - ◆実行環境のインストールがいらない!
 - ◆仮想マシンを使うJava、インタプリタを使う言語より高速に動作
- ◆②マルチプラットフォームで動作
 - ◆OSやCPUによる実行環境の差をほぼ隠蔽
 - ◆同じコードをWindows/Mac/Linuxそれぞれでビルドすると、exeや実行ファイルが作られて動かせる。クロスコンパイル(Windows上でLinux用ビルド等)も可能。
- ◆③OSに依存していない
 - ◆OS固有のAPI類に頼らずに「車輪の再発明」を敢えてやっている
 - ◆バージョン等の影響を受けにくい
- ◆④ガベージコレクターをサポート
 - ◆Cの宿命であるメモリ操作の問題を防止。「Better C」と言われる
- ◆⑤並行処理がある
 - ◆ゴルーチン、チャネルという機能で並行処理がシンプルに書ける

4. Go言語の特徴からくる→嬉しいところ



◆①圧倒的な実行速度

- ◆C/C++よりコンパイルの速度も速いとのこと
- ◆LL系言語(Ruby/Python/PHP)より10~100倍高速と言われる
- ◆→パフォーマンスやメモリを気にせず書いても大丈夫

◆②ポータビリティが高くインストール(デプロイ)が楽

- ◆OSを気にせず、実行ファイル(+設定ファイル等)をコピーするだけ
- ◆JVMや.NETのインストール、Webサーバーの準備などがいらぬ。依存関係もない
- ◆→Dockerコンテナ、Kubernetes、クラウド上で特に使いやすい

◆③標準ツール、標準ライブラリが"Battery included"で揃っている

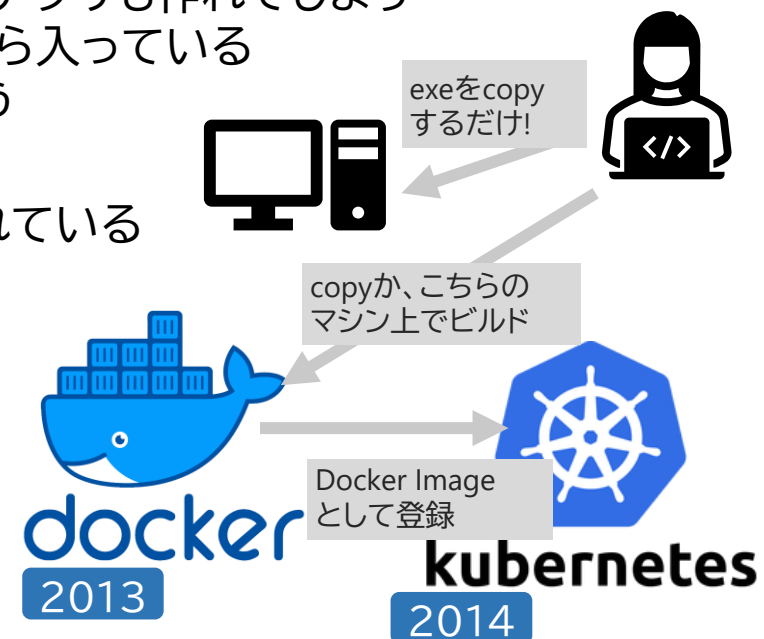
- ◆JSON、net/httpなど。フレームワークに頼らずWebアプリも作れてしまう
- ◆テストの仕組み、パッケージ管理の仕組みも最初から入っている
- ◆→Go本体さえ入れれば一通りのことができてしまう

◆④言語仕様が敢えてシンプル

- ◆構文が少なく、他言語経験者も学びやすいと言われている
- ◆→コーディング時にあまり迷わない

◆⑤コードスタイルも決まっている

- ◆go fmtという公式ツールが既に組み込まれている
- ◆→コーディング標準が不要。宗教戦争を回避



5. Go言語の使いどころと思われるシーン

◆GolangでGo!!

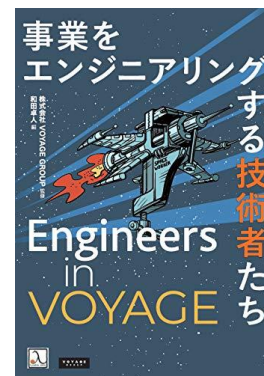
- ◆軽量級言語で作ったが速度が足りないWebアプリの置き換え
- ◆速度がいるバックエンド処理、バッチ処理
- ◆大量データの処理や演算、並列処理、基盤など
- ◆Web API系開発(APIサーバー、モックサーバー)
- ◆Docker、Kubernetes、クラウド周り、マイクロサービス
- ◆社内に展開するツール(環境の差を吸収できる)
- ◆コマンドから実行するCLIツール、ちょっとした開発ツール類
- ◆C/C++が担っていた低レイヤーの処理

◆ふつう

- ◆画面のあるアプリ(一応ライブラリはあるがあまり事例なし?)

◆使わない?

- ◆フロントエンド(ブラウザの上では動かない)
- ◆スマホのネイティブアプリ
(言語仕様上はいちおう対応)



6. 主要言語内での立ち位置①

登場年
(基本はv1)

JVM言語(Scala, Kotlin)、動的言語でPerlは省略しています

Webアプリ

フロントエンド
(クライアントサイド)

バックエンド(サーバーサイド,
バッチ系も大まかに含む)

コンパイル型言語(静的)

C#:MSファミリーの技術で揃えた際に。Webアプリではシェアが意外と少ない?
Java:伝統的でシェアも大きい。枯れている。近年はレガシーと評されることが多い...

1995

2014

CSS

HTML

JS

TS

3

5

JavaScript/TypeScript
フロントエンドの華。ブラウザ
の上で動く言語はこれだけ

2013

2015

2012



React



Vue.js



Angular

2019

WA

WebAssembly(WASM)という技術
が将来実用化されると、他の言語もブ
ラウザ上で動くようになるかも?

1996



Java

2002



Microsoft
.NET

2015



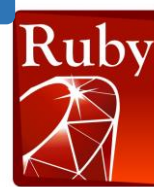
2012



インタープリタ型言語
(動的,LL,軽量級,スクリプト系)

- Pythonの開発速度とCの実行速度
- 型安全と型推論
- 大規模開発に耐えるスケーラビリティ

1995



1995



1994



python

2009



2000年代終わり頃から手軽さや開発速度、アジャイルとの親和性で若い企業で採用。
2010年代中～後半から、型の重要性、実行速度、設計の重要性などが認識され始めた。

7. 主要言語内での立ち位置②



モバイル、
スマートデバイス

ネイティブアプリ開発

モバイルバックエンド

2014



Swift

iOS: Objective-C → Swift
Android: Java → Kotlin

2016



Kotlin



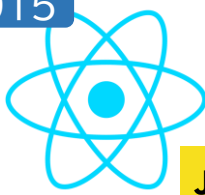
Goも仕様上はiOS、
AndroidOSに対応とあ
るが、主はバックエンド

2018



Flutter

2015



React Native

クロスプラットフォーム開発

現在はFlutter
が若干優勢

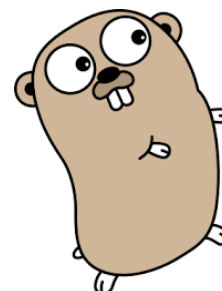
2013



Dart

JS

JS



低レイヤー、低水準、システムプログラミングの世界

コンパイル型言語(静的)

1972



1983



長らくC、C++の独壇場...

2015



2012



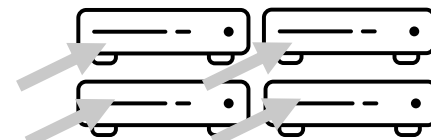
メモリセーフな開発可能に
Go: "Better C" 止まり
Rust: Cの完全置き換え
になれるかも?



8. 主要言語内での立ち位置③



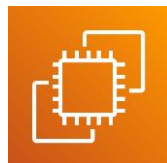
Auto scaling



exeをcopyするだけ!

クラウド
プラットフォーム

①インスタンス上



Amazon EC2

どんな言語製のアプリも載せられるのでGoもOK。デプロイしやすさが強み



②サーバーレス



AWS Lambda

対応はJava、Go、PowerShell、Node.js(JS)、C#、Python、Ruby
シェア9割がPythonとNode.js。速度が必要な場合や技術力アピールでGo採用事例。



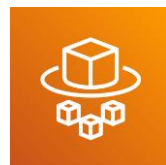
EKS



ECR



ECS



Fargate

コンテナ系サービスや
マイクロサービス系：
同じく、コンテナへの
デプロイしやすが強み

Docker/
Kubernetes



コンテナ、クラウド、マイクロ
サービスの文脈全般で、
Go言語がよく登場する。

2010



Azure
Functions

対応はC#、JavaScript、F#、Java、
PowerShell、Python、TypeScript
カスタムランタイムでGoとRustも

9. Go言語の言語仕様上のあれこれ

◆言語仕様がシンプル:「Goに入ってはGoに従え」

- ◆ `var name string = "aa"` 形式、`name := "aa"` だと型推論してくれる
- ◆ 行末のセミicolon(;)不要
- ◆ whileループがなくforループだけ
- ◆ switch文の分岐の中のbreakがいらない
- ◆ `result = score > 80 ? "OK" : "NG"` のような三項演算子がない
- ◆ if文の中が1文だったら{}省略可、がない
- ◆ 関数だけでクラスがない(!)
- ◆ オブジェクト指向の継承がない(!) 実はオブジェクト指向言語ではない。
- ◆ try-catchの例外機構がない(!) deferという機能で満たしている
- ◆ 配列は長さを宣言したら変更不可(エラー防止) 別にスライスという機能あり
- ◆ データの構造は構造体(struct)というものを使う。関数を紐づけられる
- ◆ 関数の戻り値が複数返せる。最後の戻り値で正常/異常を返すのがお作法
- ◆ 関数に関数を渡したり、関数から関数を返せたりする

なおコンパイラーは厳格です…

- ・importして使わないとエラー
- ・変数宣言して使わないとエラー



無駄を排しシンプルさと合理性を追求するGoogleの思想
“small is beautiful”を始めとしたUNIX思想の継承
Goの哲学というものが背後に見えてくる。言語にも背景や思想がある。

10. Go言語の開発環境



◆公式サイトでインストーラー提供

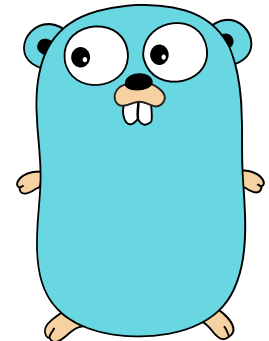
- ◆ Windows/Mac/Linuxでそれぞれバイナリが別
- ◆ あとは環境変数 `$PATH` に追加、`$GOPATH` を追加するだけ
- ◆ プロジェクト新規作成時は後から追加された **Go Modules** のコマンドを使う

環境構築は
割と簡単です

◆開発ツール

- ◆ 基本は.goのテキストファイル+αなのでツールに非依存
- ◆ 開発をサポートするツール類もコンソールベースで作られていることが多い。
- ◆ ①好きな人はvim/emacs等のテキストエディタ
- ◆ ②専用IDE: JetBrains社のGoLand (ただし有償)
- ◆ ③総合エディタ: デファクトの **Visual Studio Code(VSCode)**
 - ◆ Goプログラマの41%がVSCodeを使っているらしい(2020年記事)
 - ◆ 拡張機能の『Go』メンテがGo開発チームに。これを入れればほぼ揃う

入門には実質
VSCode一択で
よさそうです



11. 本日のまとめ

- ◆ マスコットのゴーファーくんがゆるくてかわいい。
- ◆ Webのバックエンド等で動的言語より高い性能が引き出せ、コンテナ・クラウド・マイクロサービスと親和性が高く、型の恩恵・型推論で品質確保や開発しやすさを保ち、低レイヤーの世界ではC言語の代わりにもなる言語。
→ 他の言語のシェアを奪いにとって代わるのではなく、隙間を上手く埋めた立ち位置にある言語です。
- ◆ 技術トレンド的には、コンパイル型の言語の中ではこのGoとRustが注目されています。



それではまた次回！

Creative Commons Attributions 3.0
The Gopher character is based on the Go mascot designed by Renée French.