



Golang-Korea, APR 23 2019

고등학생이 golang으로 코딩하기



Las aka Wonho

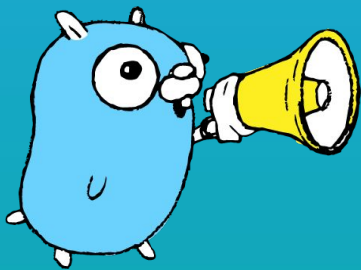
GSM

Student



Hello Golang!

Objective



Today's (glorious) blather.

Who I Am	01
Why Golang	02
How to Learn	03
Try and fail	04
In School	05
More and More	06

introduction

Who I Am?

저는

Gwangju Software Meister High School

Love OpenSource

개발자 지망생

Functional programming

Open Source Contributor

Functional Programming

Intern (Server Engineer)

직장 구하는중...

Mail : haho6629@gmail.com

Twitter : @LasWonho

SECTION ONE

Why Golang?

Youtube

Just Interest

Google Make Golang

Docker made by golang

Effective Go



Why Golang?



Gofmt

Why Golang?



```
if (a > b) {  
    c = a + b  
}
```

```
if( a > b )  
{  
    c = a + b  
}
```

```
if (a > b)  
{  
    c = a + b  
}
```

```
if (a > b)  
    c = a + b
```

I **Love** Learn New Language

Why Golang?



- Elixir
- Ruby
- Rust
- Scala
- Clojure
- And.... Golang!

개발 1년차

NOT No Base

Why Golang?



- Java
- Python
- C++
- Javascript
- Assembly
- C#(Unity)

Hello Golang!

Summary

SECTION TWO

How to Learn?

arahansa/Golkorea

📖 README.md

GolangKorea Facebook

Golang Korea facebook group 멤버들과 나누는 깃헙 일단 이런 테마를 가지고 정리를 하고 있습니다.

1. Gotour
2. GoByExample
3. Effective Go
4. packages 샘플 소스들 정리

그룹 내의 발표자료들.

(그룹 홈페이지 잘 정리해서 저장할 예정입니다.)

1. Go, 90% Perfect 100% of the time
2. Advanced Go Concurrency Patterns
3. Go + Revel +Gorp 로 만드는 페이지징 게시판
4. Go + Revel+ GORM 으로 만드는 블로그
5. [Go 웹 프레임워크 Revel 번역] (<http://100coding.com/go/revel>)

Go Tour

나만의 자료 만들기

How to Learn?



Las-Wonho/GO



📖 구조체 접근.go	줄정리	4 months ago
📖 구조체.go	줄정리	4 months ago
📖 강릉 배열.go	줄정리	4 months ago
📖 다중 리턴 함수.go	Go Lint	4 months ago
📖 라이브러리 사용하기.go	줄정리	4 months ago
📖 맵.go	줄정리	4 months ago
📖 맵으로 놀기.go	줄정리	4 months ago
📖 메소드.go	Go Lint	4 months ago
📖 문자열.go	Update 문자열.go	4 months ago
📖 반복.go	줄정리	4 months ago
📖 배열 기초.go	줄정리	4 months ago
📖 배열 다루기 1.go	줄정리	4 months ago
📖 범위 기반 반복.go	줄정리	4 months ago
📖 상수 숫자.go	줄정리	4 months ago
📖 상수.go	줄정리	4 months ago
📖 서버.go	줄정리	4 months ago
📖 소켓서버_0.go	Update 소켓서버_0.go	4 months ago
📖 소켓서버_1.go	Create 소켓서버_1.go	4 months ago
📖 시스템 정보 확인.go	줄정리	4 months ago
📖 요상한 자료형.go	줄정리	4 months ago
📖 인터페이스.go	줄정리	4 months ago
📖 일단 변수부터.go	줄정리	4 months ago
📖 조건문 두번째.go	줄정리	4 months ago
📖 조건문 첫번째.go	줄정리	4 months ago
📖 출력하기.go	줄정리	4 months ago
📖 불려져.go	줄정리	4 months ago
📖 포인터.go	줄정리	4 months ago
📖 함수 리턴값 이름 정해놓기.go	줄정리	4 months ago
📖 헬로 월드.go	줄정리	4 months ago

Problem Solving



Golang을 배우자!



wonho ha

May 23, 2018 · 3 min read

Go라는 언어를 아시나요? 약간 C와 비슷하면서 특이한 현대언어입니다. 구글에서 만든 언어로 유명하죠. 정적 언어이며 단순하고 깔끔한 문법을 가진 현대언어입니다. 컴파일 속도도 빠르며 런타임의 성능 또한 보장되어있습니다. 그러나 현대적인 부분이 많이 없다고 느낄수도 있지만, 현대언어입니다. Rust와 많이 비교됩니다. 실제로 Rust가 더 빠릅니다. 그러나 동시성 처리를 할 때,



SECTION THREE

Try and fail

Goland



VS Code + VIM

Golang in Assembly

gccgo

Setting up and using gccgo

Releases	Imports
Source code	Debugging
Building	C Interoperability
Gold	Types
Prerequisites	Function names
Build commands	Automatic generation of Go declarations from C source code
Using gccgo	
Options	

This document explains how to use gccgo, a compiler for the Go language. The gccgo compiler is a new frontend for GCC, the widely used GNU compiler. Although the frontend itself is under a BSD-style license, gccgo is normally used as part of GCC and is then covered by the [GNU General Public License](#) (the license covers gccgo itself as part of GCC; it does not cover code generated by gccgo).

Note that gccgo is not the `gc` compiler; see the [Installing Go](#) instructions for that compiler.

Try and Fail



Example

```
$ gccgo -o main main.go
$ main // Error... main not found

$ ./main
Hello World~!
```

Linux 속

Functional Golang

- Pointer
- Goroutine, Channel
- Generic
- Tail Call Optimization

Example

```
router := httprouter.New()
router.GET("/", Index)
router.GET("/A", func (w http.ResponseWriter, r *http.Request, _ httprouter.Params){
    fmt.Fprint(w, "<h1>Las World!</h1>\n")
    fmt.Fprint(w, "", arr)
})
router.GET("/A:X", func (w http.ResponseWriter, r *http.Request, ps httprouter.Params){
    I := ps.ByName("X")
    X, _ := strconv.Atoi(I[0:2])
    Y, _ := strconv.Atoi(I[2:4])
    V, _ := strconv.Atoi(I[4:])
    arr[X][Y]=V
    fmt.Fprint(w, "X :", X, "\nY :", Y, "\nV :", V)
})
log.Fatal(http.ListenAndServe(":"+strconv.Itoa(8080), router))
```

High Order Function


Hello Golang!

Summary

SECTION FOUR

In School

GSM



- OS
- C/C++/C#
- Java
- Network System
- AI
- Web Programming
- Database
- Software Engineering
- Android Programming

C++ Class

AHeui Interpreter

aheui/goaheui

CppHeui

App Programming Class

Godot - Share Your Dot Cavus

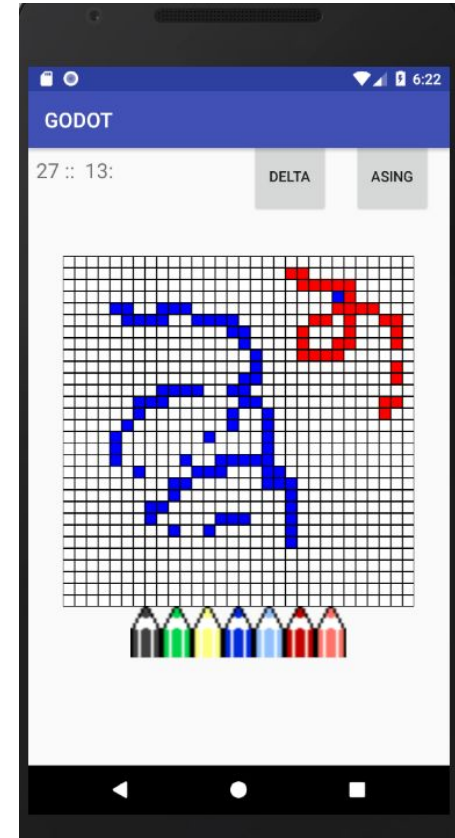
GoToServer

School Project

Develop for App

Using Goroutine

Http api server



main.go

```
func help(){
    fmt.Println("      Go To Server      ")
    fmt.Println(">This Program is Wonho's Server Controller")
    fmt.Println(">Used Golang, Source is https://github.com/Las-Wonho/GoToServer")
    fmt.Println(">Server Port was 8080")
    fmt.Println(">>Command list<<")
    fmt.Println("    ->help")

    fmt.Println("    ->state")

    fmt.Println("    ->start")

    fmt.Println("    ->version or ver")

    fmt.Println("    ->end")
}
```

Server 실행
도움말

main.go

```
func CLI_io(state chan int){
    var command string
    for {
        fmt.Print(">")
        fmt.Scanln(&command)
        Method(command, state)
    }
}

func wait(state chan int){
    for <-state != 0{
        time.Sleep(time.Second)
    }
}

func main() {
    Start()
    state := make(chan int,0)
    go CLI_io(state)
    wait(state)
    fmt.Println("System End")
}
```

Server

main.go

```
func Server(){
    var arr = [30][30]int{}
    router := httprouter.New()
    router.GET("/", Index)
    router.GET("/A", func (w http.ResponseWriter, r *http.Request, _ httprouter.Params){
        fmt.Fprint(w, "<h1>Las World!</h1>\n")
        fmt.Fprint(w, "", arr)
    })
    router.GET("/A/:X", func (w http.ResponseWriter, r *http.Request, ps httprouter.Params){
        I := ps.ByName("X")
        X, _ := strconv.Atoi(I[0:2])
        Y, _ := strconv.Atoi(I[2:4])
        V, _ := strconv.Atoi(I[4:])
        arr[X][Y]=V
        fmt.Fprint(w, "X :", X, "\nY :", Y, "\nV :", V)
    })
    log.Fatal(http.ListenAndServe(":"+strconv.Itoa(8080), router))
}
```

Server

Chat Server

SocketServer

More Goroutines

Multi User Chat

Putty로 접근 가능(TCP)

Now Available Service

main.go

```
func sender(pipe *chan string, count *int) {  
    for {  
        data := <-*pipe  
        for i, e := range connections {  
            if e == nil {  
                continue  
            }  
            if i > *count {  
                break  
            }  
            sendData(e, data+"\r\n")  
        }  
    }  
}
```

Server

SECTION FIVE

More and More

gin

HTTP3

devsisters/goquic

lucas-clemente/quic-go

Go2

More and More



proposal: Go 2: unsafe: add Assume or Unreachable Go2 NeedsDecision Proposal	19
<small>#30582 opened on 5 Mar by josharian Proposal</small>	
proposal: Go 2: range types Go2 LanguageChange NeedsInvestigation Proposal	3
<small>#30428 opened on 27 Feb by beoran Proposal</small>	
proposal: Go 2: let := support any l-value that = supports Go2 LanguageChange NeedsDecision Proposal	29
<small>#30318 opened on 20 Feb by bradfitz Proposal</small>	
proposal: Go 2: spec: add integer types with explicit overflow behavior, and remove unchecked operations on built-in integers Go2 LanguageChange Proposal	21
<small>#30209 opened on 14 Feb by bcmills Proposal</small>	
proposal: Go 2: ban calling append with a single argument Go2 NeedsDecision Proposal	21
<small>#30040 opened on 1 Feb by kaedys Proposal</small>	
proposal: Go 2 error values Go2 NeedsInvestigation Proposal Proposal-Accepted	248
<small>#29934 opened on 25 Jan by jba Go1.13</small>	
proposal: Go 2: sync/atomic: type modifiers as an addition to types Go2 LanguageChange NeedsInvestigation Proposal	3
<small>#29501 opened on 2 Jan by tema3210 Proposal</small>	
proposal: Go 2: remove dot imports from the language Go2 LanguageChange NeedsDecision Proposal	32
<small>#29326 opened on 19 Dec 2018 by ianlancetaylor Proposal</small>	

SECTION END

END