

Go 기본 문법 - 1

컴소과 동아리 REFERENCE

제작자: 김명준

목 차

- Go 언어란?
- 변수와 상수
- 조건문과 연산자
- 반복문
- 함수

Go 언어란?

GO LANG?

구글에서 개발한 프로그래밍 언어

Go 언어는 간결하고 빠른 속도를 지향하며,
병렬 처리와 동시성(concurrency)을 지원



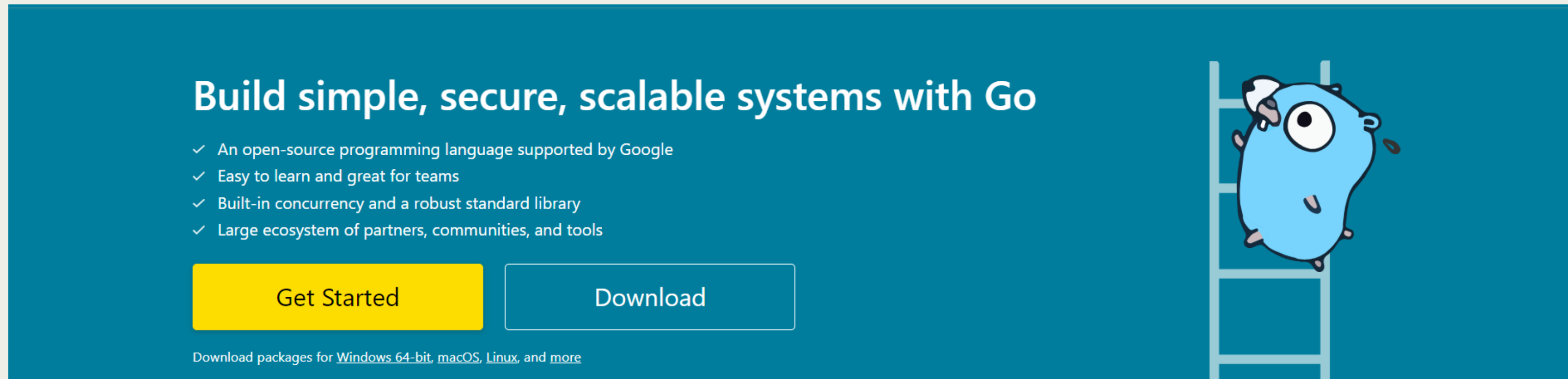
구글에서 개발된 Kubernetes 및 Docker,
Etcd 등의 컨테이너 오케스트레이션
(orchestration) 시스템에서 사용

G O 의 특징

- 간결한 문법
- 멀티 쓰레드 지원
- 메모리 관리
- 컴파일 언어
- 크로스 플랫폼 지원

GO 다운로드

Go 홈페이지: <https://go.dev/>

A screenshot of the Go landing page. The background is a solid teal color. On the left, the text "Build simple, secure, scalable systems with Go" is written in white. Below this, there are four bullet points, each preceded by a white checkmark. At the bottom left, there are two buttons: a yellow one labeled "Get Started" and a white one with a teal border labeled "Download". Below the "Download" button, there is a line of small white text. On the right side of the banner, there is a cartoon illustration of a blue, blob-like creature with large eyes and a small mouth, standing on a white ladder.

Build simple, secure, scalable systems with Go


- ✓ An open-source programming language supported by Google
- ✓ Easy to learn and great for teams
- ✓ Built-in concurrency and a robust standard library
- ✓ Large ecosystem of partners, communities, and tools

[Get Started](#) [Download](#)

Download packages for [Windows 64-bit](#), [macOS](#), [Linux](#), and [more](#)

Download 클릭

GO 다운로드

Why Go ▾LearnDocs ▾PackagesCommunity ▾

All releases






After downloading a binary release suitable for your system, please follow the [installation instructions](#).

If you are building from source, follow the [source installation instructions](#).

See the [release history](#) for more information about Go releases.

As of Go 1.13, the go command by default downloads and authenticates modules using the Go module mirror and Go checksum database run by Google. See <https://proxy.golang.org/privacy> for privacy information about these services and the [go command documentation](#) for configuration details including how to disable the use of these servers or use different ones.

Featured downloads

Microsoft Windows Windows 10 or later, Intel 64-bit processor  go1.21.1.windows-amd64.msi	Apple macOS (ARM64) macOS 11 or later, Apple 64-bit processor  go1.21.1.darwin-arm64.pkg	Apple macOS (x86-64) macOS 10.15 or later, Intel 64-bit processor  go1.21.1.darwin-amd64.pkg	Linux Linux 2.6.32 or later, Intel 64-bit processor  go1.21.1.linux-amd64.tar.gz	Source  go1.21.1.src.tar.gz
---	--	--	--	--

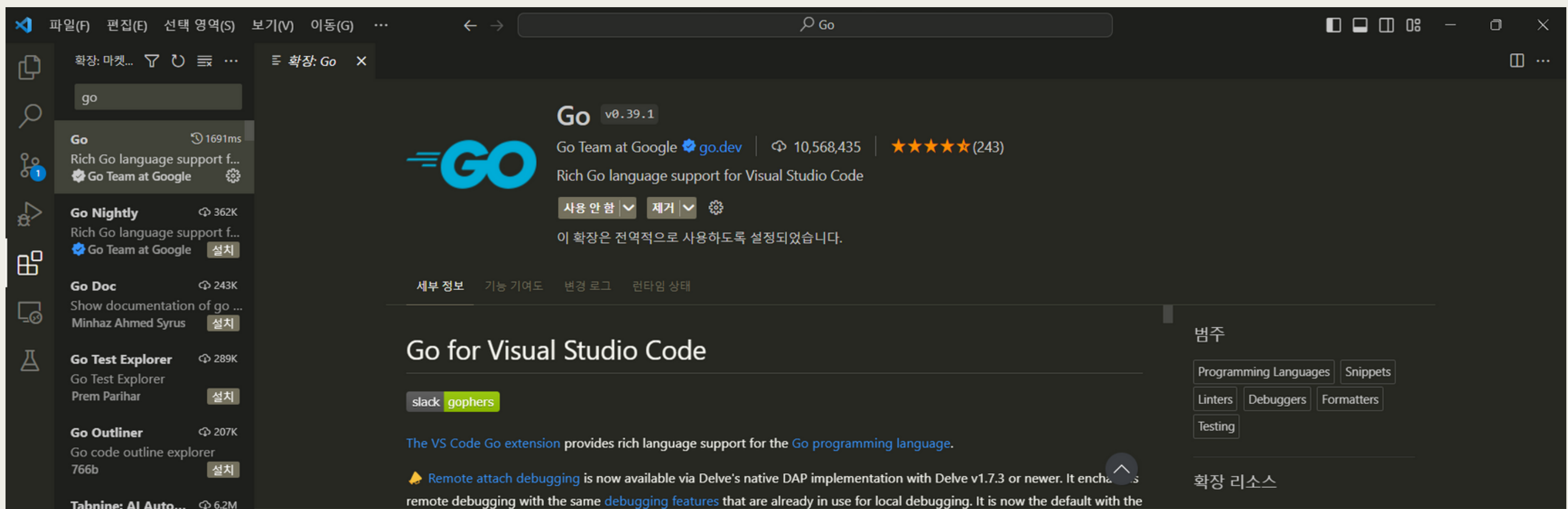
Stable versions

go1.21.1 ▾

OS에 맞게 설치 파일 다운로드

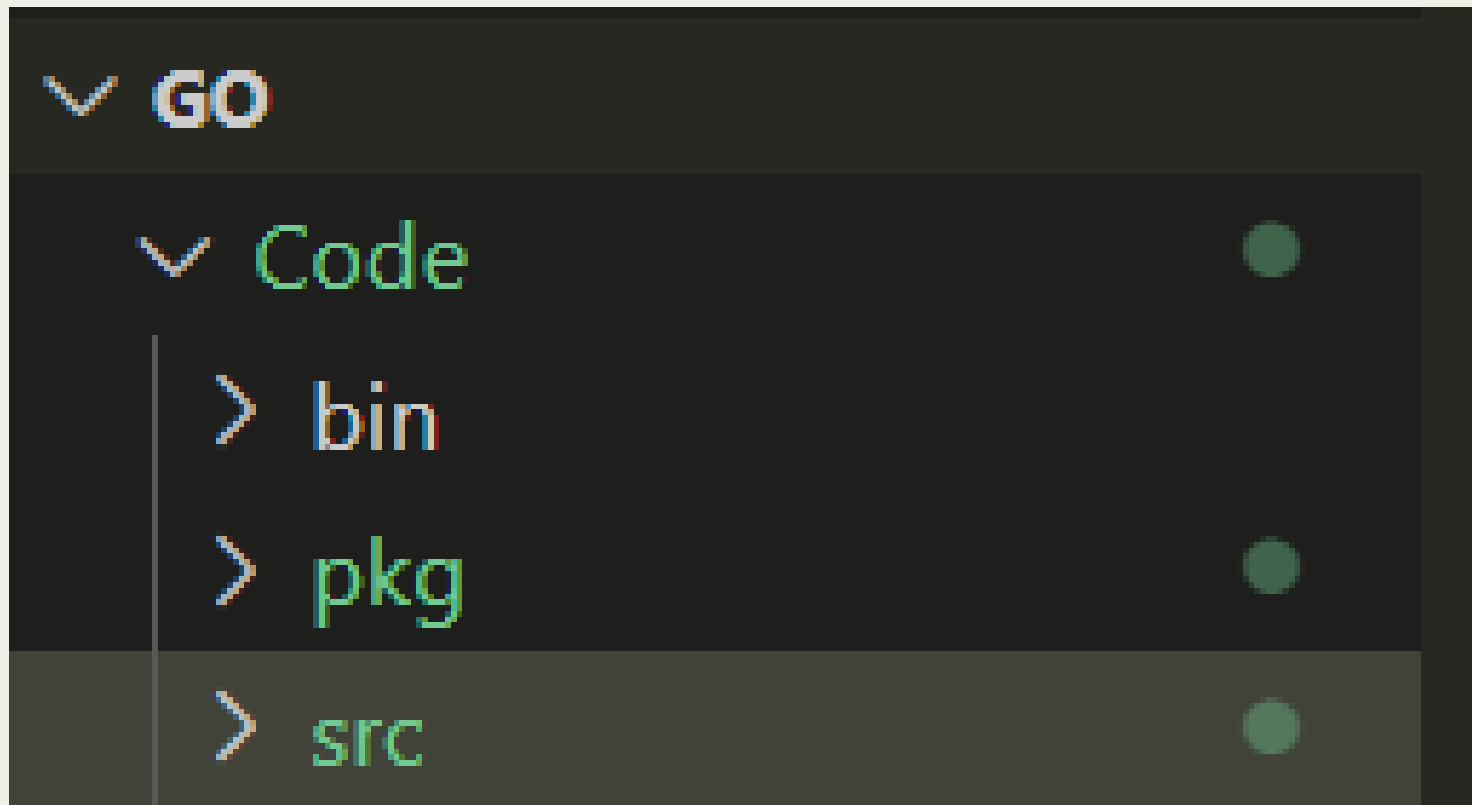
GO 다운로드

설치가 완료됐다면 VS CODE의 확장으로 들어가 Go 검색



Go 설치

GO 다운로드



프로젝트를 만들고

반드시 bin, pkg, src

폴더를 생성

그 후에 환경 변수에 새로 만들기

GOPATH

D:\Git_Repository\Go\Code

GOPATH에 프로젝트 경로를 입력

GO 다운로드

bin: 소스 파일 컴파일 후 운영체제별 실행 가능한 바이너리 파일이 저장되는 곳

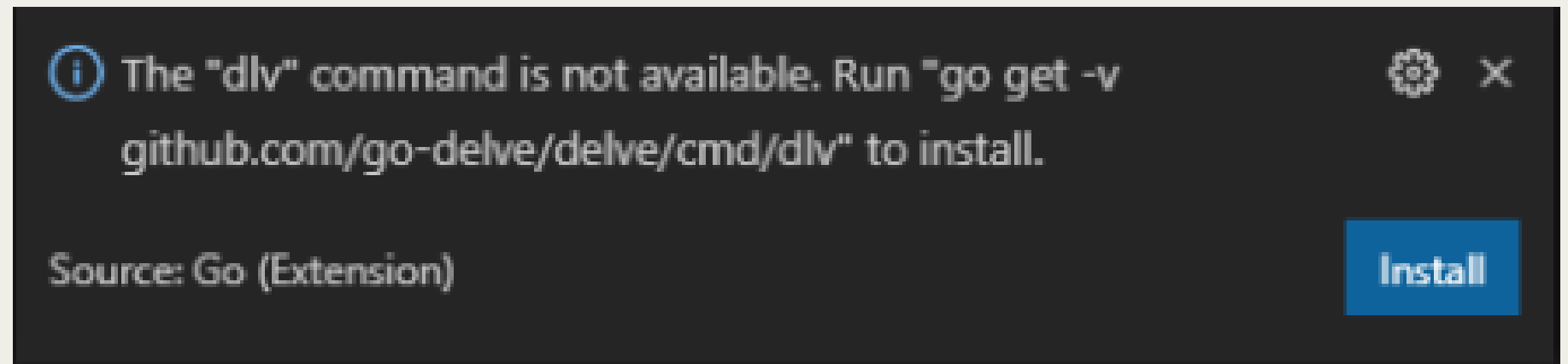
pkg: 프로젝트에 필요한 패키지가 컴파일 되어 라이브러리 파일이 저장되는 곳

src: 직접 작성한 소스 코드 및 오픈 소스 코드를 저장하는 곳

GO 다운로드

```
GO Hello.go U X
GO Hello.go > ...
1 package main
2
3 func main() {
4     println("Hello Wolrd")
5 }
6
```

간단하게 Hello World를 출력하는
Hello.go 파일을 만든 후 실행하면



이 창이 뜰 건데 install 누름

GO 다운로드

다운로드 다 한 후에

```
문제   출력   디버그 콘솔   터미널   포트

myoun@Mangjun MINGW64 /d/Git_Repository/Go (main)
$ go run Hello.go
Hello wolrd
```

터미널에서

go run Hello.go를 입력하면 실행

변수와 상수

데이터 타입

- 논리형: bool
- 문자형: string
- 정수형: int[8, 16, 32, 64], uint[8, 16, 32, 64], uintptr
- 실수형: float32, float64, complex64, complex128
- 기타: byte(바이트 코드), rune(유니 코드)

변수

Go에선 변수를 **var** 키워드를 사용하여 선언

1. `var i int = 1`

할당되는 값을 보고 타입을 추론 가능해서 생략 가능

2. `var i = 1`

`:=` 를 사용해 선언(`:=`는 함수 내에서만 가능)

3. `i := 1`

변수를 선언하고 초기화하지 않으면 **Zero Value**를 할당

숫자형: 0

string: "" (빈 문자열)

bool: false

상수

Go에선 상수를 **const** 키워드를 사용하여 선언

할당되는 값을 보고 타입을 추론 가능해서 생략 가능

1. `const i int = 1`

2. `const i = 1`

여러 개의 상수를 묶어서 지정할 때는 괄호()를 사용

```
const (  
    name = "김명준"  
    age  = 25  
)
```

```
const (  
    i = iota  
    ii  
    iii  
)
```

값을 `iota`로 주면 상수값을 0부터 순차대로 할당

조건문과 연산자

연산자

C언어와 똑같다

산술 연산자

관계 연산자

논리 연산자

비트 연산자

할당 연산자

포인터 연산자(포인터 산술은 제공 X)

조건문 - IF문

if문은 다른 언어와 유사함

if문의 조건식은 괄호로 감싸지 않아도 됨

그러나 조건 블록 시작 브레이스({)는

반드시 if문과 같은 라인에 있어야 함

조건문 - SWITCH문

Go 언어에서 switch는 다른 언어와 유사

다른 점

- switch 문의 표현식 생략 가능(true로 인식)
- case 문에 if문처럼 조건식이 들어갈 수 있음
- case 문은 break를 생략해도 항상 break함(fallthrough 사용)
- case 문을 변수의 값 대신에 타입으로 나눌 수 있음

반복문

반복문

Go에서는 반복문이 **for** 하나뿐

for 초기값; 조건식; 증감 {}의 형식(for 문 뒤에 괄호()를 쓰면 에러)

무한 반복

```
for {  
    println("Hello world")  
}
```

반복문

컬렉션에서 쓰는 for range문

for 인덱스, 요소값 := range 컬렉션 {}의 형식

```
names := []string{"김", "명", "준"}  
for index, name := range names {  
    println(index, name)  
}
```

함수

함수

Go에선 함수를 **func** 키워드를 통해 정의

다른 언어와는 다르게 매개변수를 **변수 타입** 순으로 작성

```
func printA(msg string) {  
    print(msg)  
}
```

매개변수를 전달하는 방식은 C 언어와 같이

by Value와 by Reference(포인터) 방식이 있음

함수

가변 인자 사용

```
func main() {  
    say("김", "명", "준")  
}  
  
func say(msg ...string) {  
    for _, s := range msg {  
        print(s)  
    }  
}
```

리턴값이 있는 함수

```
func add(num1 int, num2 int) int {  
    return num1 + num2  
}
```

Go에선 타입이 뒤에 있음

함수

다른 언어와 다르게 Go에선 리턴 값이 여러 개 일 수 있음

```
func sum(num ...int) (int, int) {  
    sum := 0  
    count := 0  
    for _, n := range num {  
        sum += n  
        count++  
    }  
  
    return sum, count  
}
```

리턴 타입을 괄호()로 묶어줌

```
func sum(num ...int) (sum int, count int) {  
    for _, n := range num {  
        sum += n  
        count++  
    }  
  
    return  
}
```

반환 값에 이름을 줄 수도 있음

(return문 생략 불가)

감사합니다

컴소과 동아리 REFERENCE