



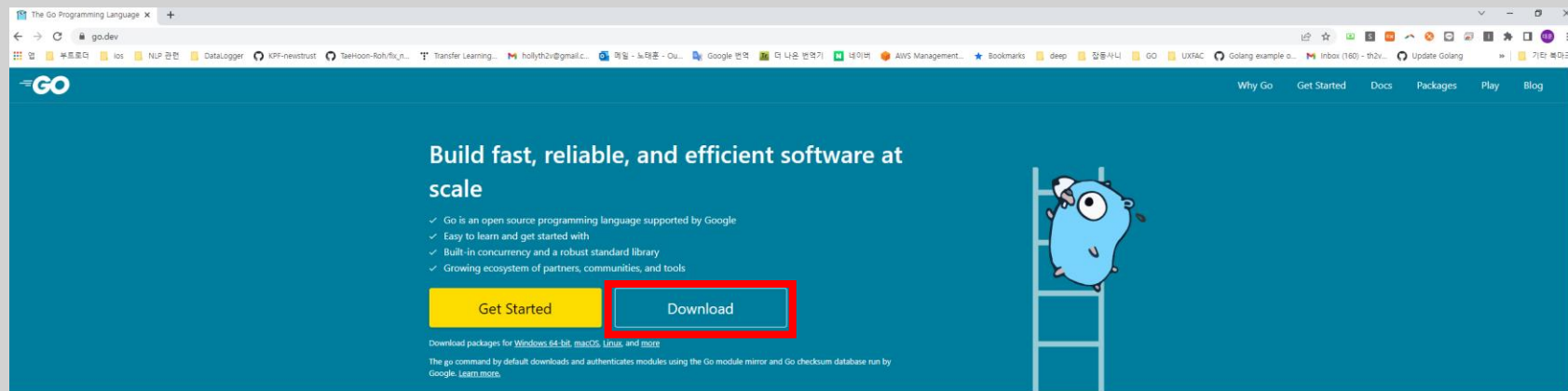
핵심! Go 프로그래밍

• 1일차

목차

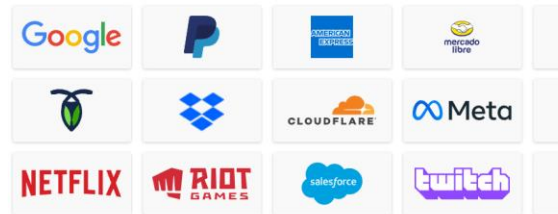
- Golang 설치
- Vscode 설치
- Git 설치

Golang 설치



Companies using Go

Organizations in every industry use Go to power their software and services [View](#)



Downloads

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 7 or later, Intel 64-bit processor
[go1.18.2.windows-amd64.msi](#)
(130MB)

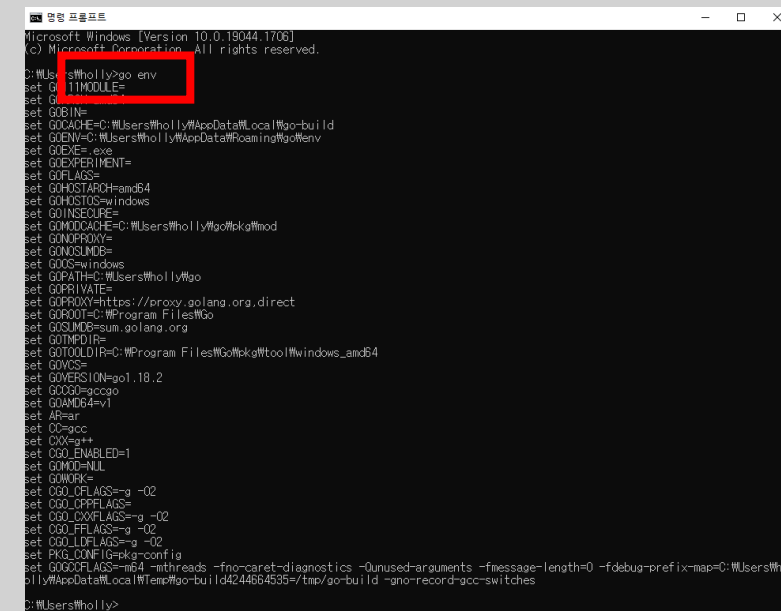
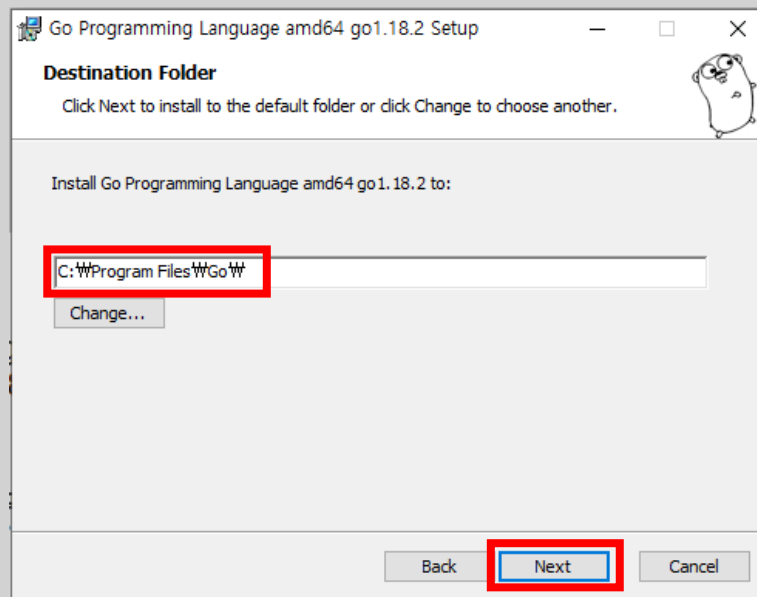
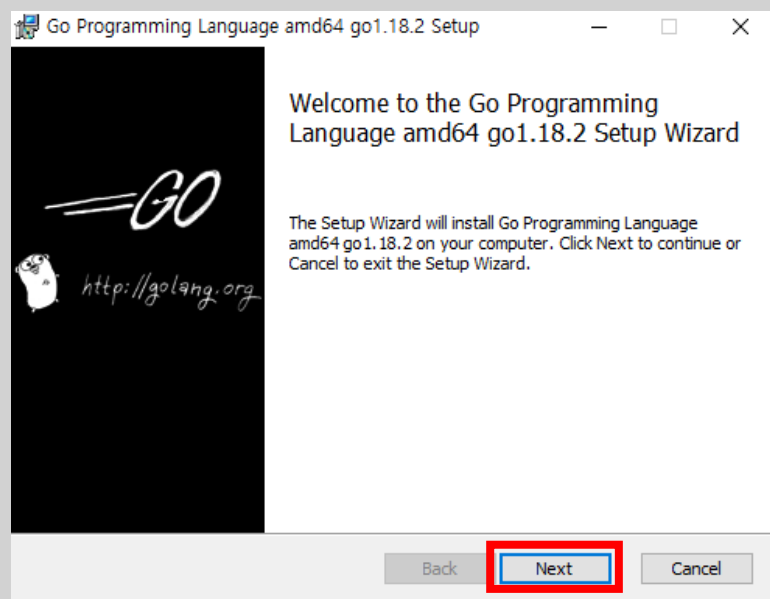
Apple macOS
macOS 11 or later, Apple 64-bit processor
[go1.18.2.darwin-arm64.pkg](#)
(132MB)

Apple macOS
macOS 10.13 or later, Intel 64-bit processor
[go1.18.2.darwin-amd64.pkg](#)
(138MB)

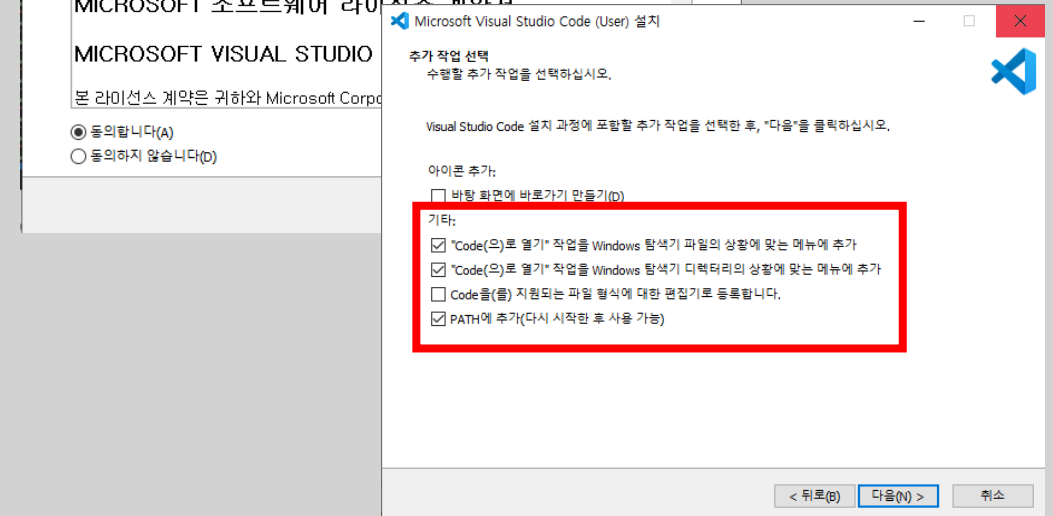
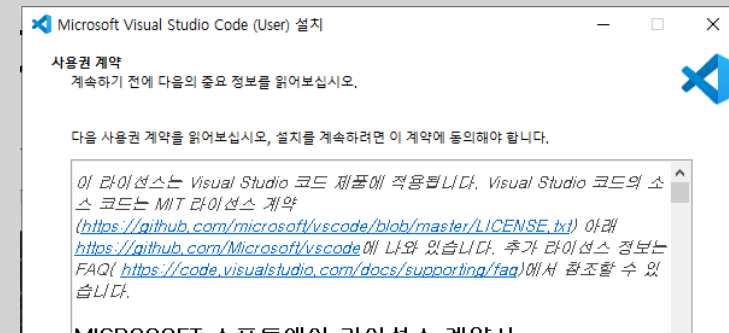
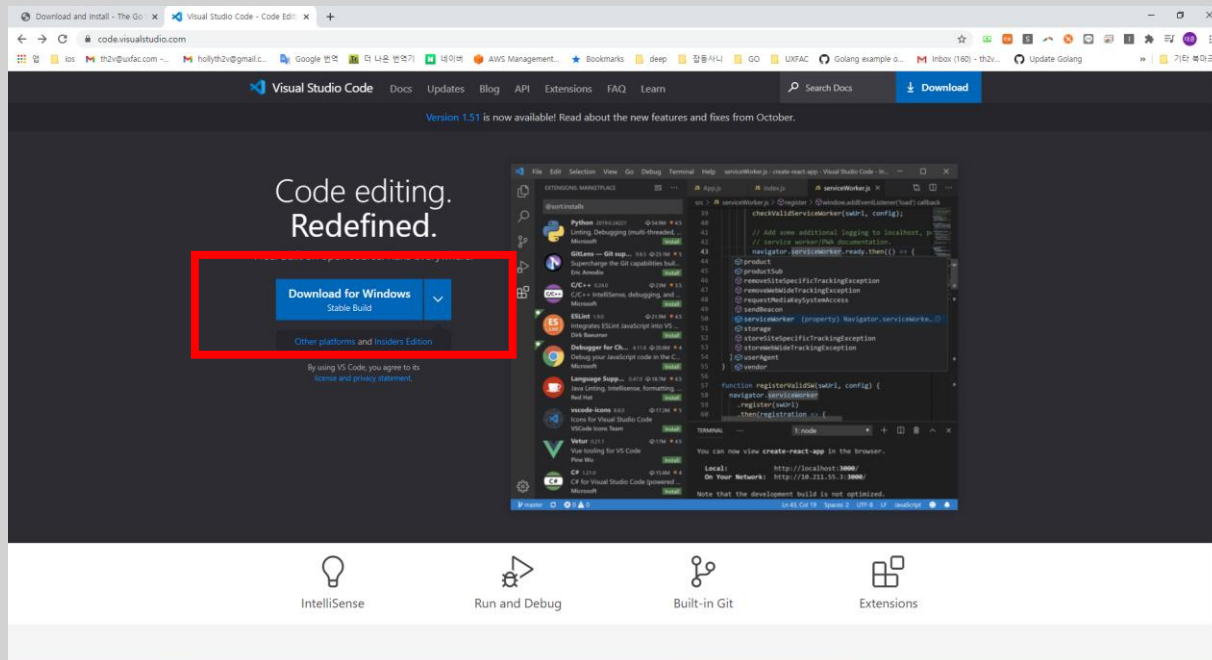
Linux
Linux 2.6.23 or later, Intel 64-bit processor
[go1.18.2.linux-amd64.tar.gz](#)
(135MB)

Source
[go1.18.2.src.tar.gz](#) (22MB)

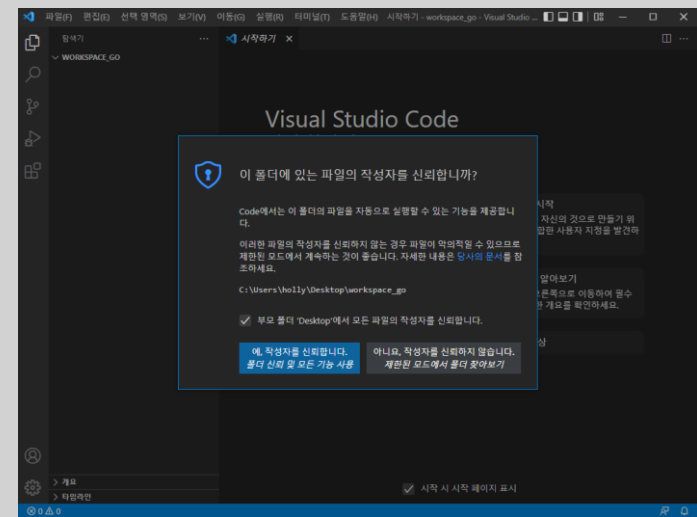
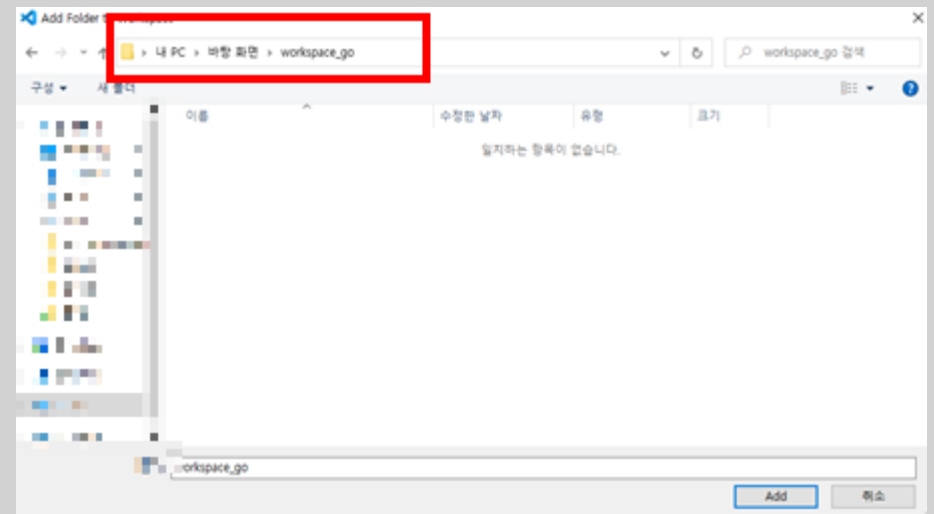
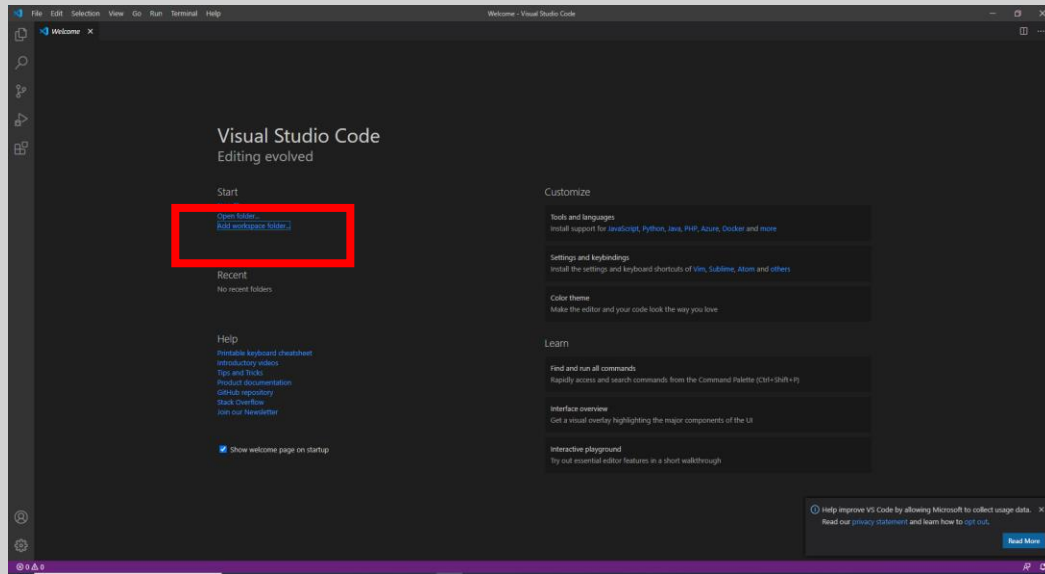
Golang 설치



Vscode 설치

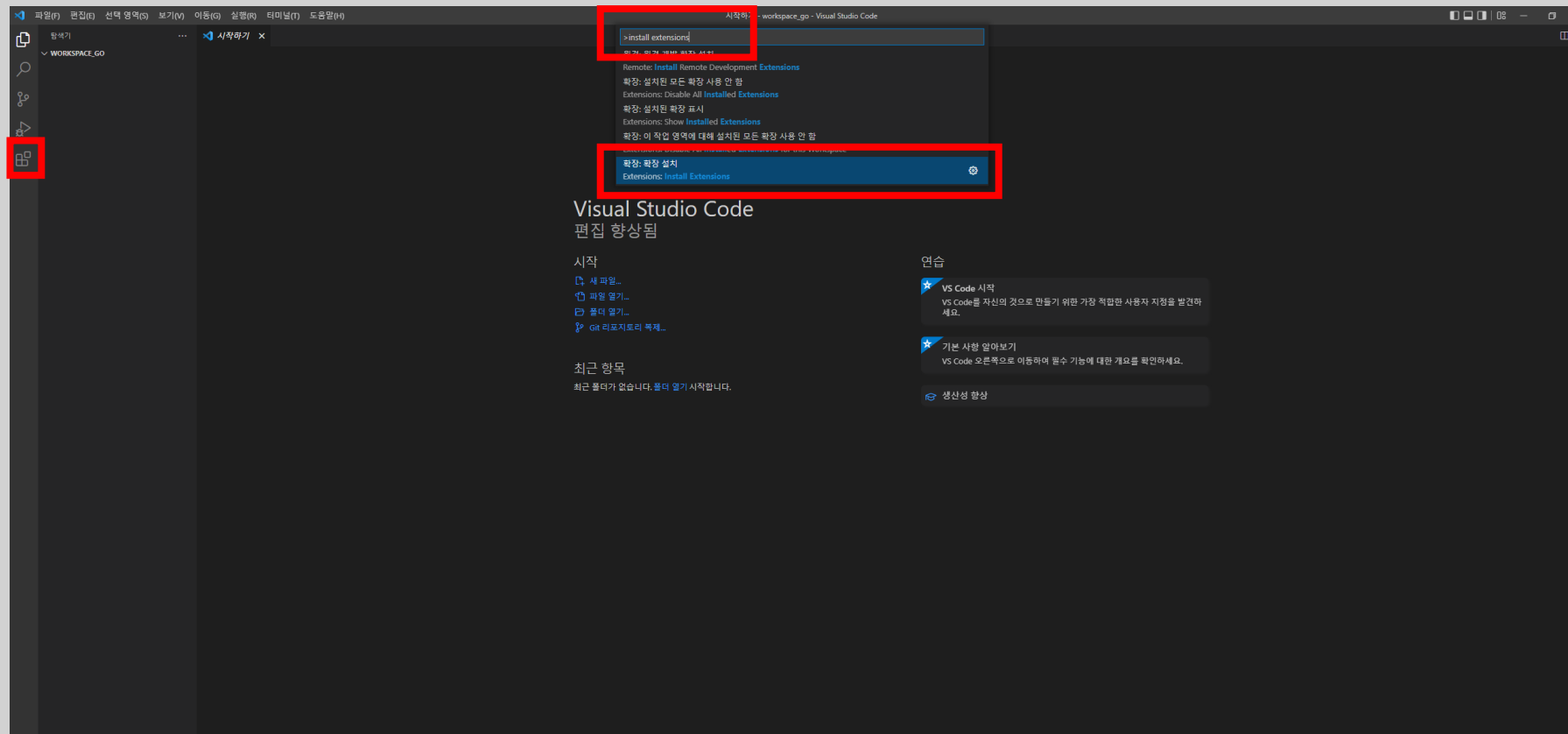


Vscode 설치



Vscode 설치

CTRL + SHIFT + P or 보기 -> 명령어 plate



Vscode 설치

확장: 마켓플레이스

Go v0.33.0

Go Team at Google | 6,984,576 | ★★★★★ (229)

Rich Go language support for Visual Studio Code

설치

세부 정보 기능 기여도 변경 로그

Go for Visual Studio Code

slack gophers

The VS Code Go extension provides rich language support for the Go programming language.

Remote attach debugging is now available via Delve's native DAP implementation with Delve v1.7.3 or newer. It enhances remote debugging with the same debugging features that are already in use for local debugging. It is now the default with the Go Nightly build of the extension and will become the default for the stable releases in mid 2022. We recommend switching your remote attach configurations in `launch.json` to use "debugAdapter": "dlv-dap" now to verify that this works for you. Please file a new issue if you encounter any problems.

Watch Debugging Treasure Hunt from GopherCon 2021 for a fun take on a debugging demo with VS Code Go and Delve DAP.

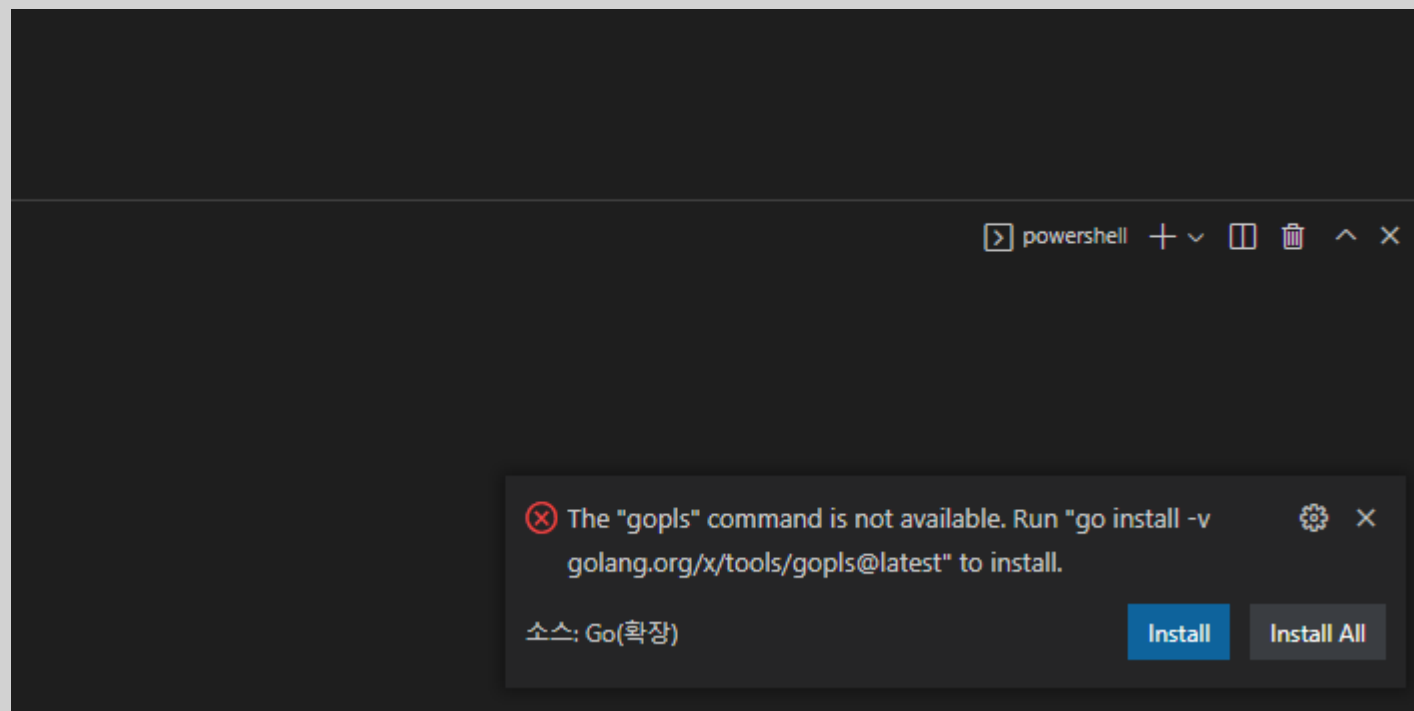
Quick Start

Welcome! 🍷

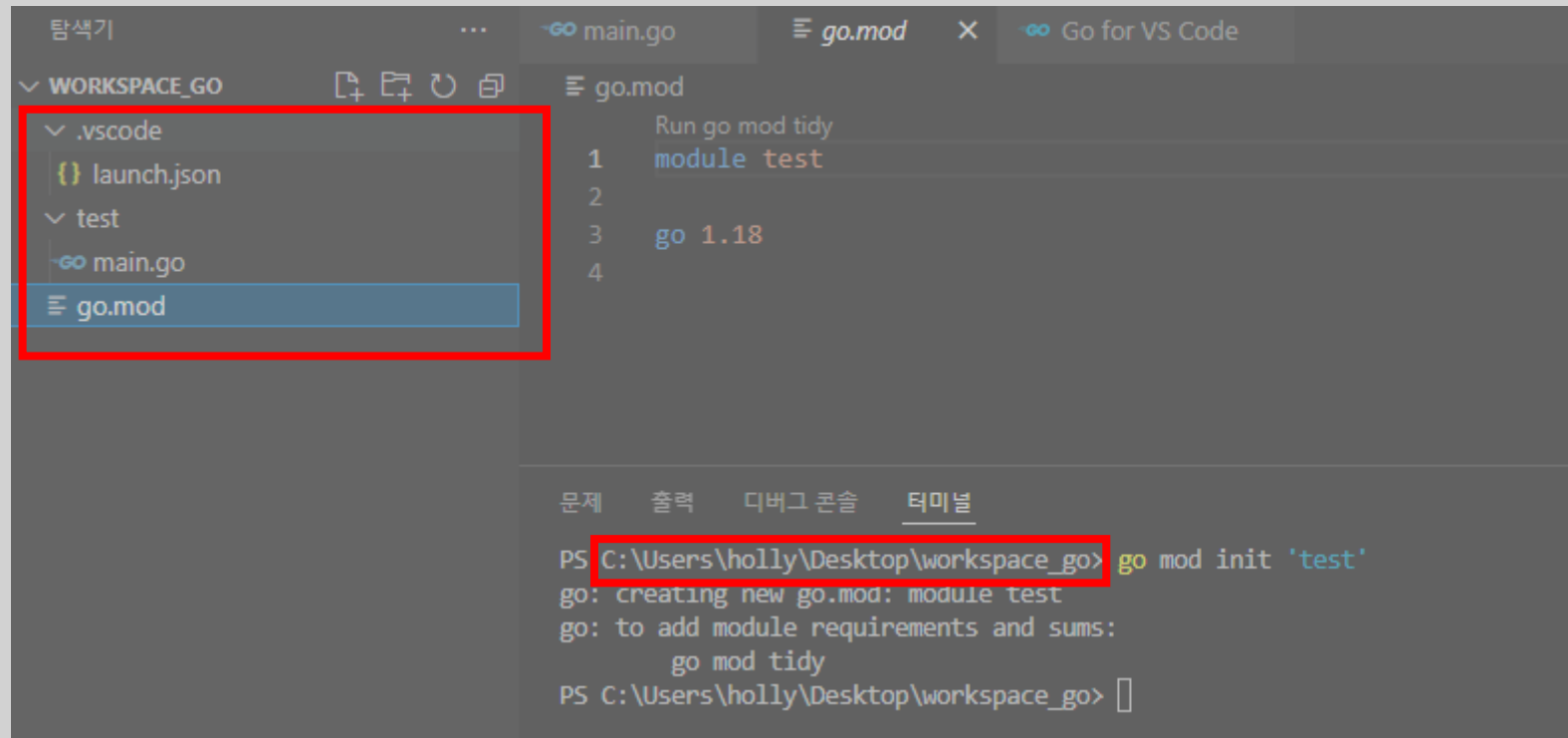
Whether you are new to Go or an experienced Go developer, we hope this extension fits your needs and enhances your development experience.

- Step 1. If you haven't done so already, install Go and the VS Code Go extension.
 - Go installation guide. This extension works best with Go 1.14+.
 - Managing extensions in VS Code.
- Step 2. To activate the extension, open any directory or workspace containing Go code. Once activated, the Go status bar will appear in the bottom left corner of the window and show the recognized Go version.
- Step 3. The extension depends on a set of extra command-line tools. If they are missing, the extension will show the "Analysis Tools Missing" warning. Click the notification to complete the installation.

Vscode 설치

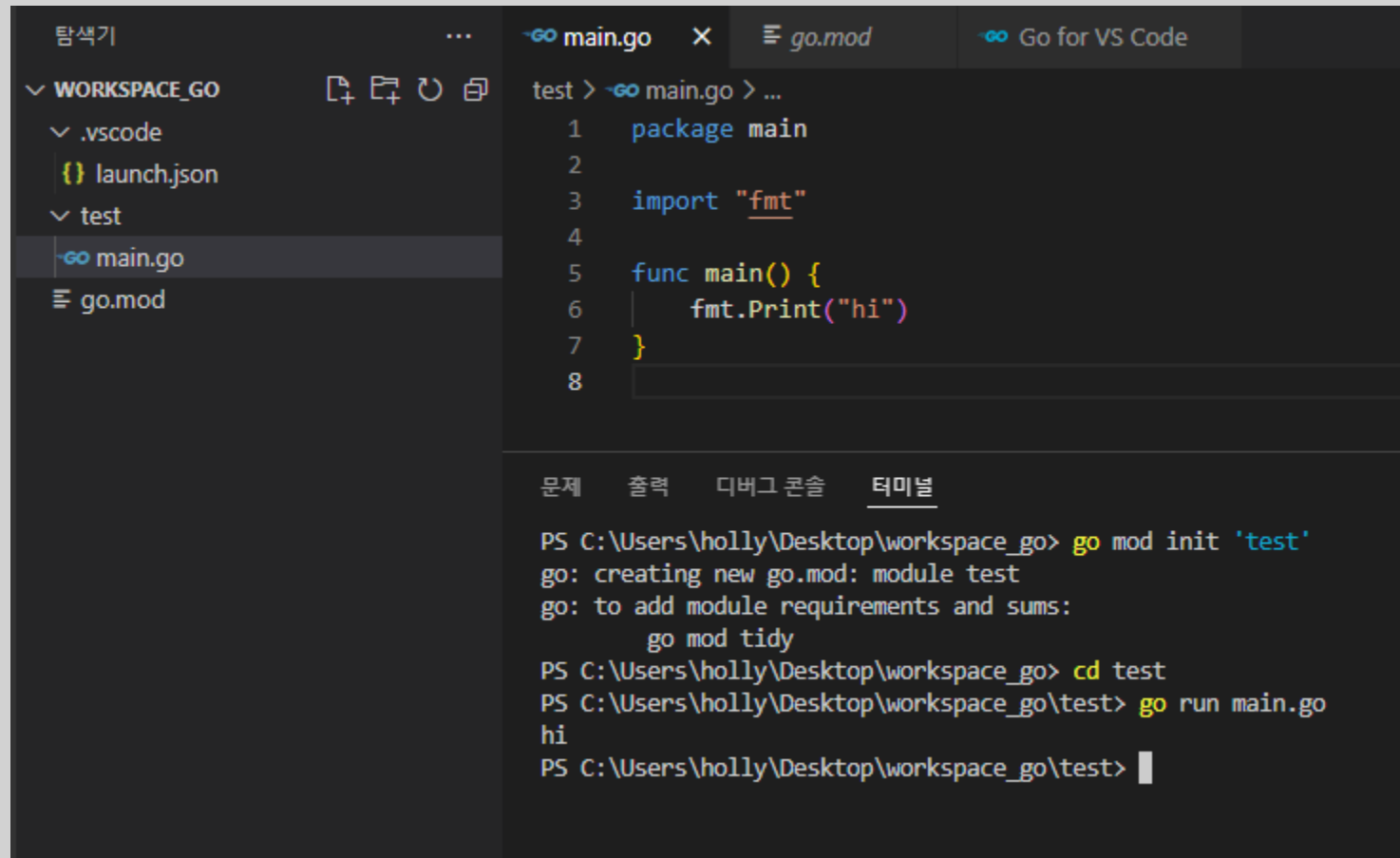


Vscode 설치



Root Dir에서 go mod init 'package name'

Vscode 설치



The screenshot displays the VS Code interface with a Go project. The left sidebar shows the file explorer with the following structure:

- WORKSPACE_GO
 - .vscode
 - launch.json
 - test
 - main.go
 - go.mod

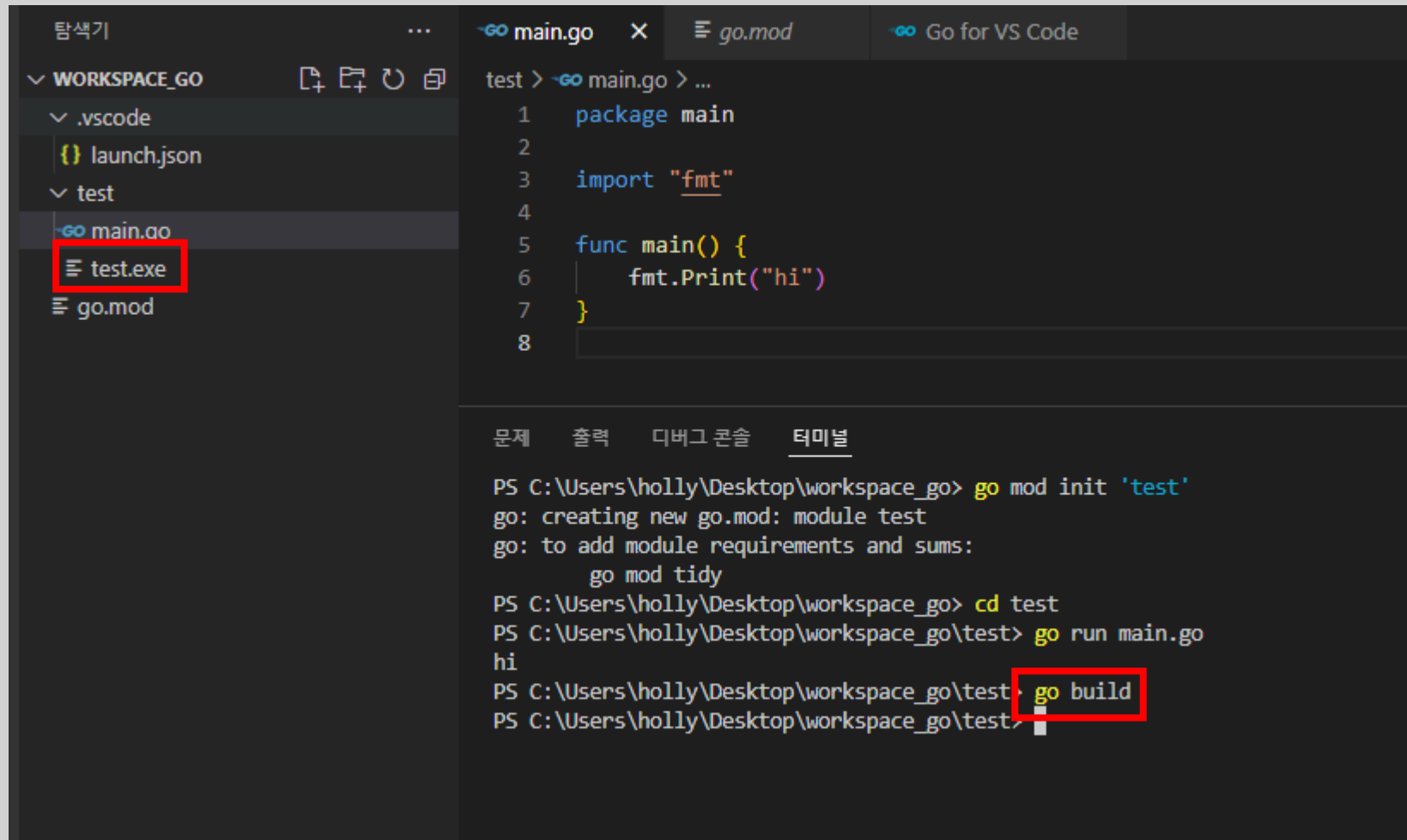
The main editor area shows the `main.go` file with the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Print("hi")
7 }
8
```

The bottom panel shows the terminal output for the command `go run main.go`:

```
PS C:\Users\holly\Desktop\workspace_go> go mod init 'test'
go: creating new go.mod: module test
go: to add module requirements and sums:
    go mod tidy
PS C:\Users\holly\Desktop\workspace_go> cd test
PS C:\Users\holly\Desktop\workspace_go\test> go run main.go
hi
PS C:\Users\holly\Desktop\workspace_go\test>
```

Vscode 설치



The screenshot shows the VS Code interface with a Go project. The Explorer sidebar on the left shows the file structure: `WORKSPACE_GO` (expanded) contains `.vscode` (with `launch.json`), `test` (expanded), `main.go`, `test.exe` (highlighted with a red box), and `go.mod`. The Editor pane shows `main.go` with the following code:

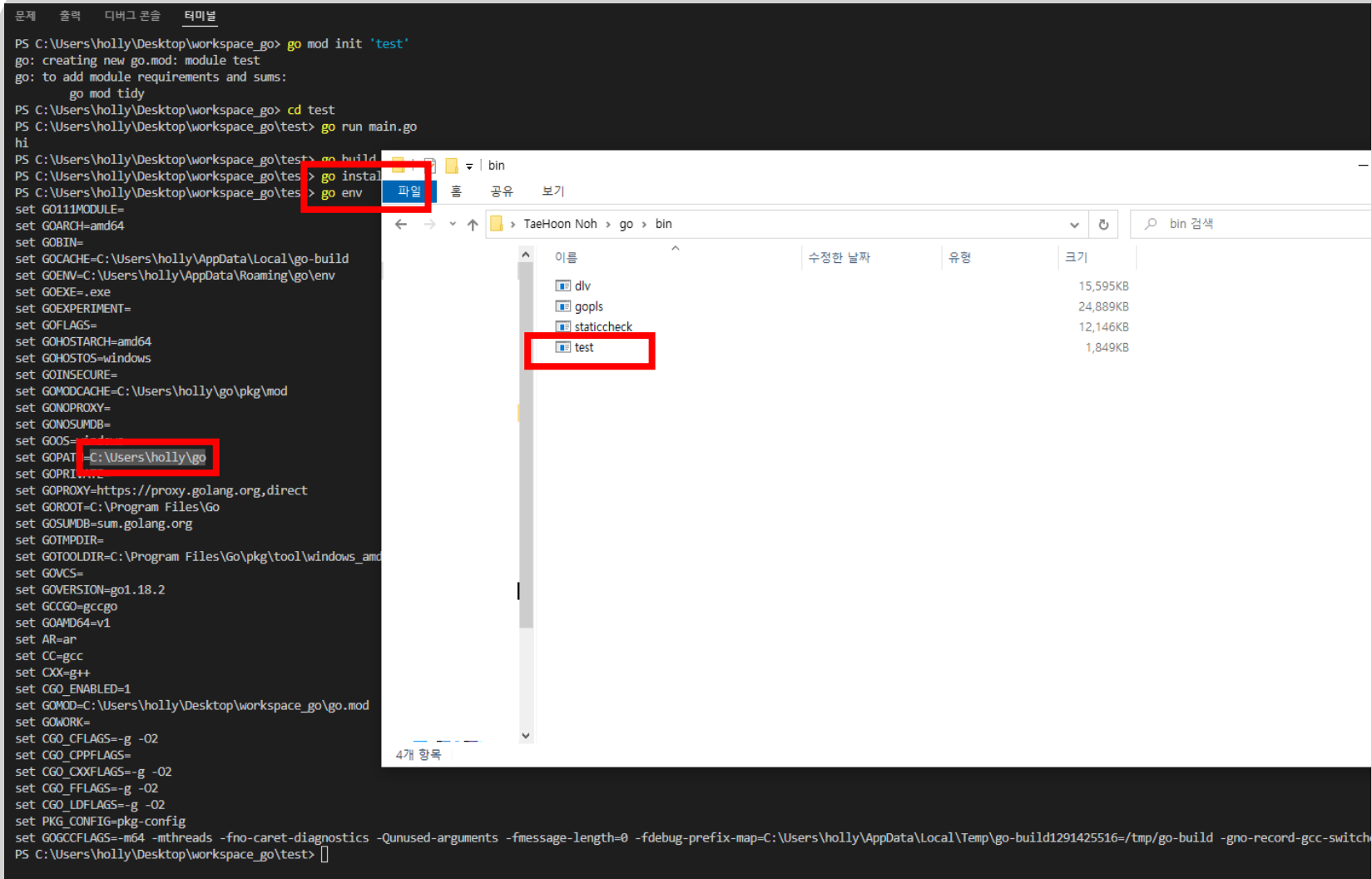
```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Print("hi")
7 }
8
```

The Terminal pane at the bottom shows the command prompt output:

```
PS C:\Users\holly\Desktop\workspace_go> go mod init 'test'
go: creating new go.mod: module test
go: to add module requirements and sums:
    go mod tidy
PS C:\Users\holly\Desktop\workspace_go> cd test
PS C:\Users\holly\Desktop\workspace_go\test> go run main.go
hi
PS C:\Users\holly\Desktop\workspace_go\test> go build
PS C:\Users\holly\Desktop\workspace_go\test>
```

The `go build` command in the terminal is highlighted with a red box.

Vscode 설치



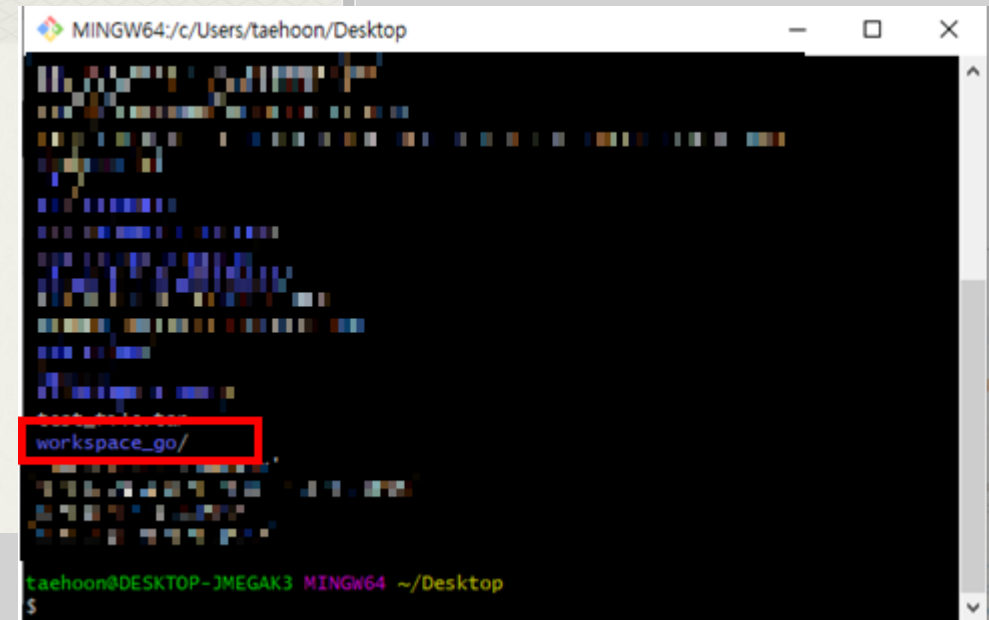
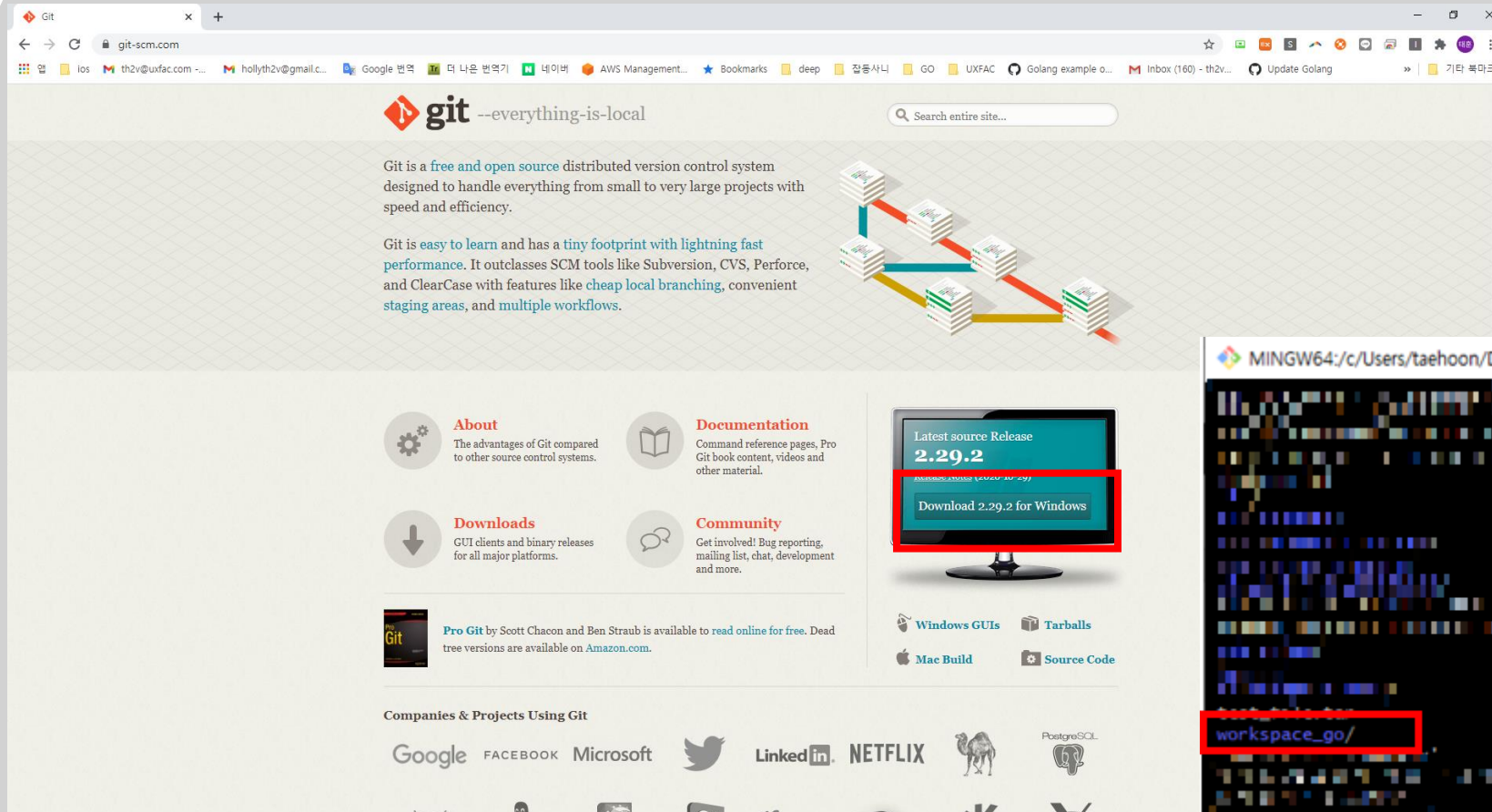
The image shows a VS Code window with a terminal and a file explorer. The terminal displays the output of Go commands for initializing a module and setting environment variables. The file explorer shows the contents of the `bin` directory, which includes executables like `dlv`, `gopls`, `staticcheck`, and `test`.

```
PS C:\Users\holly\Desktop\workspace_go> go mod init 'test'
go: creating new go.mod: module test
go: to add module requirements and sums:
    go mod tidy
PS C:\Users\holly\Desktop\workspace_go> cd test
PS C:\Users\holly\Desktop\workspace_go\test> go run main.go
hi
PS C:\Users\holly\Desktop\workspace_go\test> go build
PS C:\Users\holly\Desktop\workspace_go\test> go install
PS C:\Users\holly\Desktop\workspace_go\test> go env
set GO111MODULE=
set GOARCH=amd64
set GOBIN=
set GOCACHE=C:\Users\holly\AppData\Local\go-build
set GOENV=C:\Users\holly\AppData\Roaming\go\env
set GOEXE=.exe
set GOEXPERIMENT=
set GOFLAGS=
set GOHOSTARCH=amd64
set GOHOSTOS=windows
set GOINSECURE=
set GOMODCACHE=C:\Users\holly\go\pkg\mod
set GONOPROXY=
set GONOSUMDB=
set GOOS=
set GOPATH=C:\Users\holly\go
set GOPRIVATE=
set GOPROXY=https://proxy.golang.org,direct
set GOROOT=C:\Program Files\Go
set GOSUMDB=sum.golang.org
set GOTMPDIR=
set GOTOOOLDIR=C:\Program Files\Go\pkg\tool\windows_amd64
set GOVCS=
set GOVERSION=go1.18.2
set GCCGO=gccgo
set GOAMD64=v1
set AR=ar
set CC=gcc
set CXX=g++
set CGO_ENABLED=1
set GOMOD=C:\Users\holly\Desktop\workspace_go\go.mod
set GOWORK=
set CGO_CFLAGS=-g -O2
set CGO_CPPFLAGS=
set CGO_CXXFLAGS=-g -O2
set CGO_FFLAGS=-g -O2
set CGO_LDFLAGS=-g -O2
set PKG_CONFIG=pkg-config
set GOGCCFLAGS=-m64 -mthreads -fno-caret-diagnostics -Qunused-arguments -fmessage-length=0 -fdebug-prefix-map=C:\Users\holly\AppData\Local\Temp\go-build1291425516=/tmp/go-build -gno-record-gcc-switches
PS C:\Users\holly\Desktop\workspace_go\test>
```

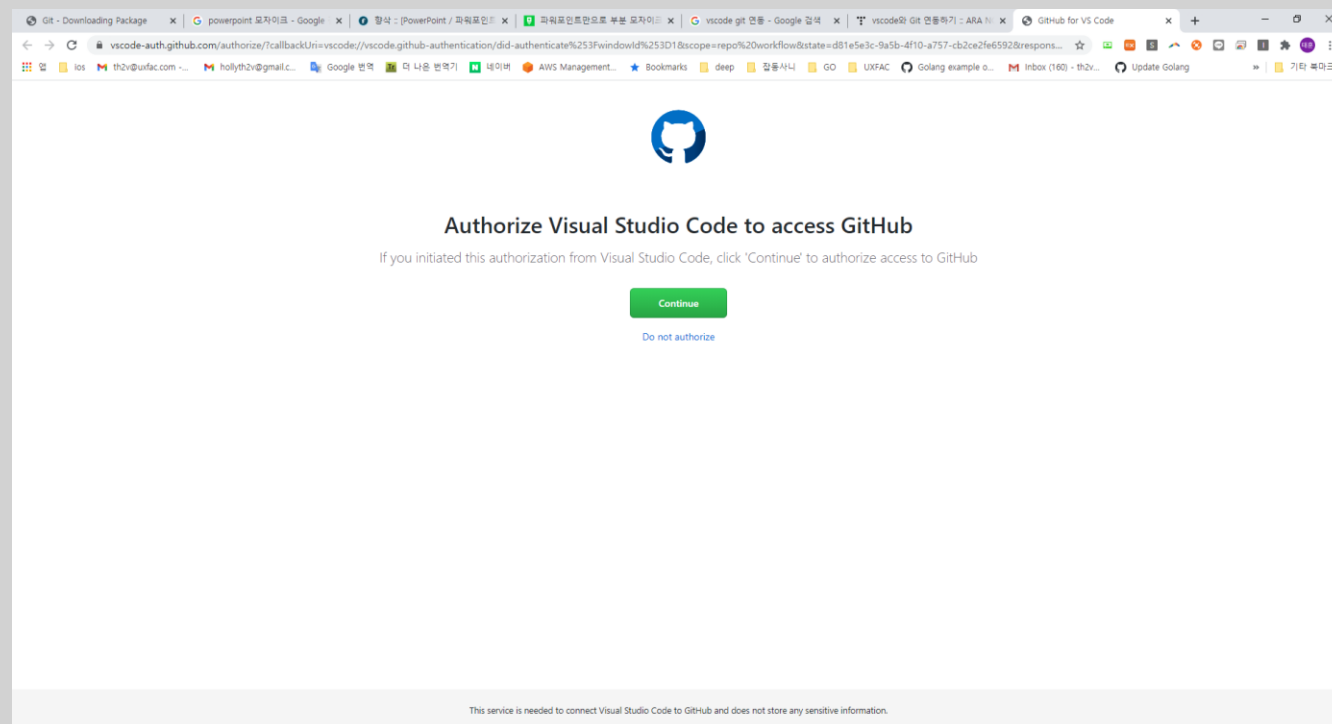
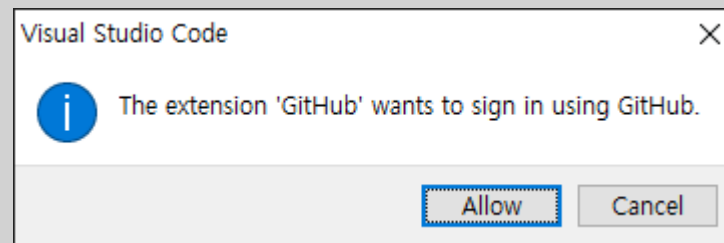
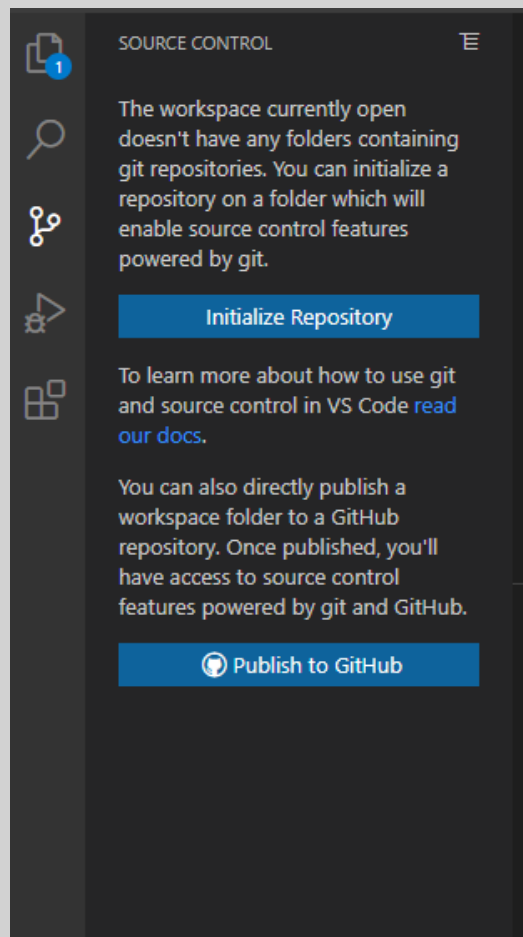
File Explorer (C:\Users\holly\Desktop\workspace_go\test\bin):

이름	수정한 날짜	유형	크기
dlv			15,595KB
gopls			24,889KB
staticcheck			12,146KB
test			1,849KB

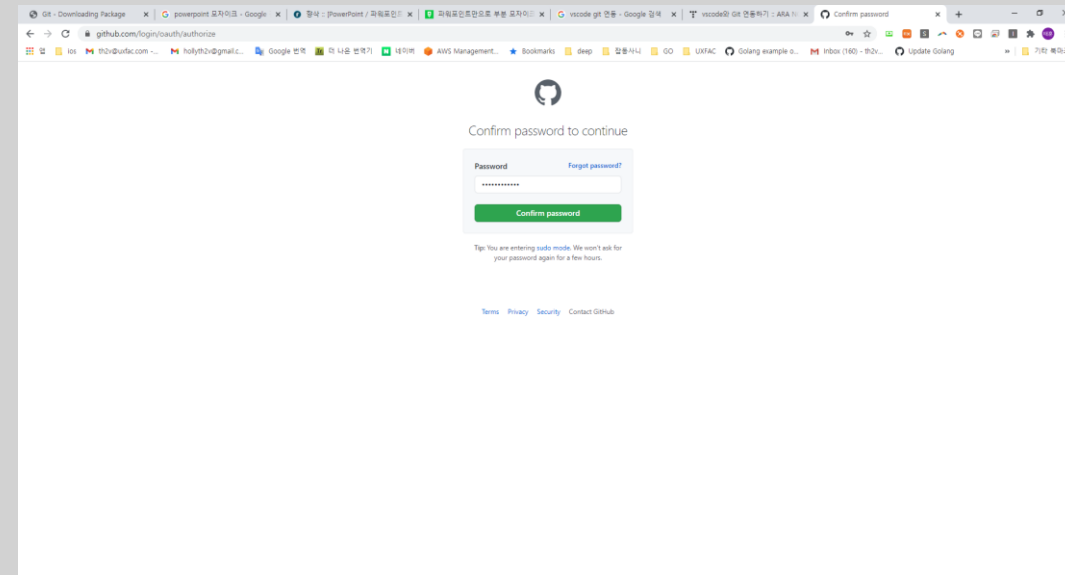
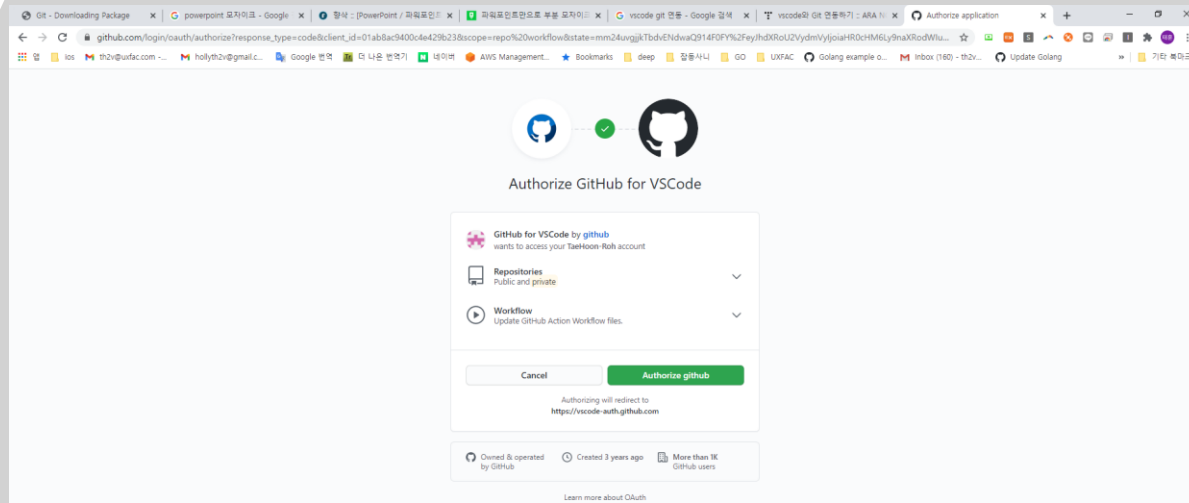
Git 설치



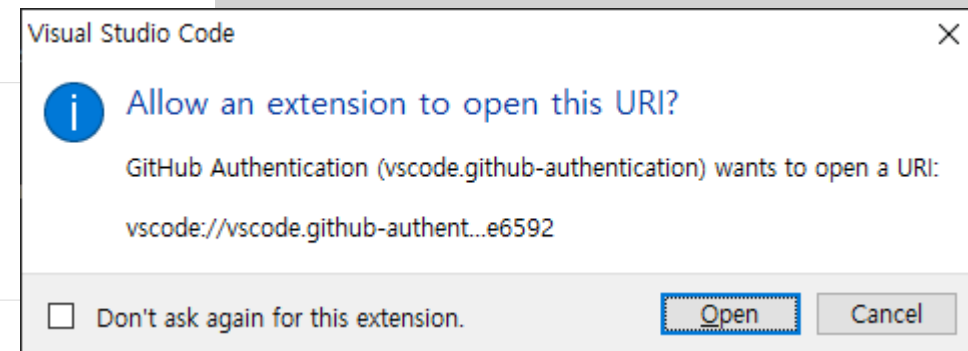
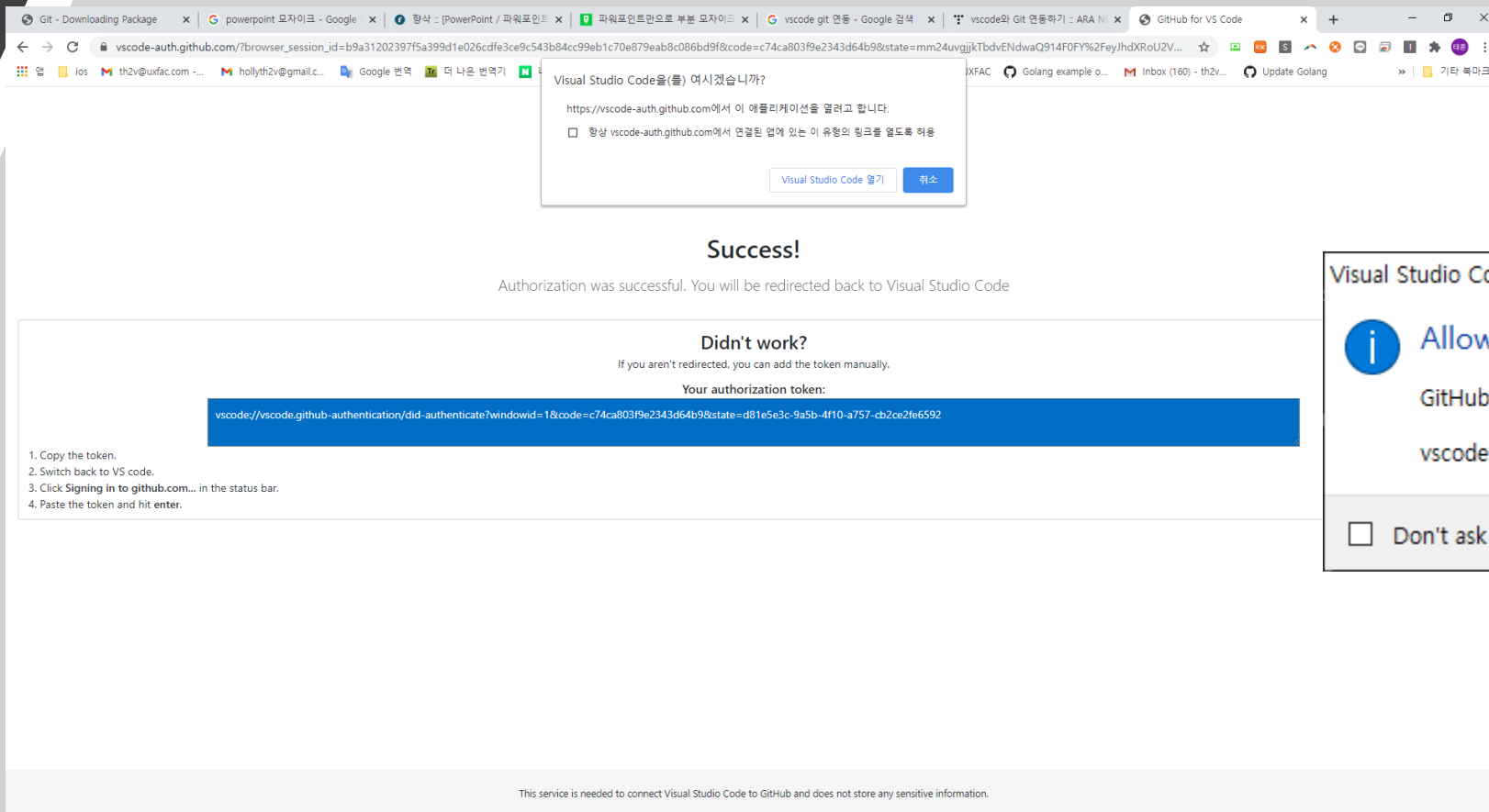
Git 설치



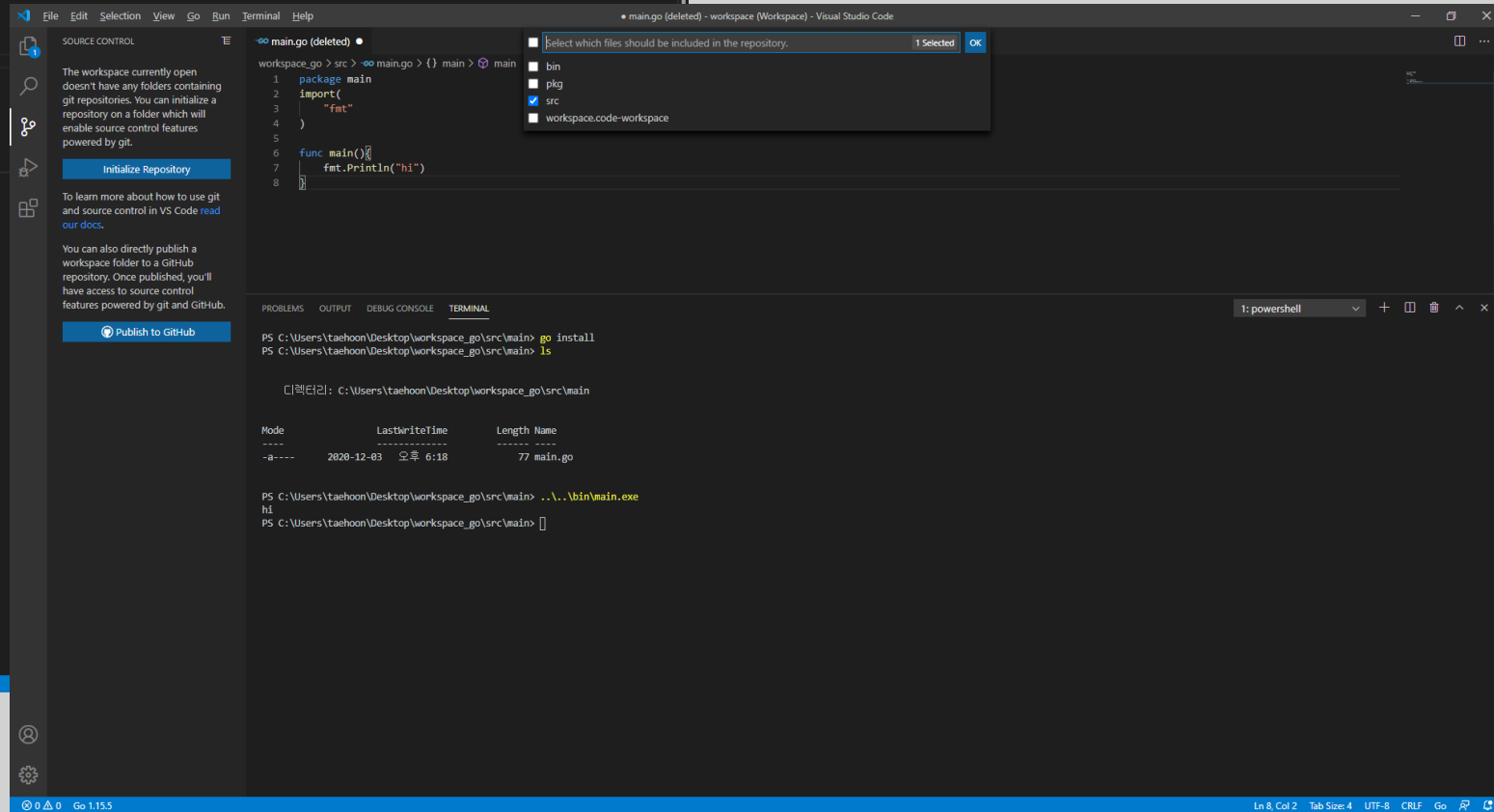
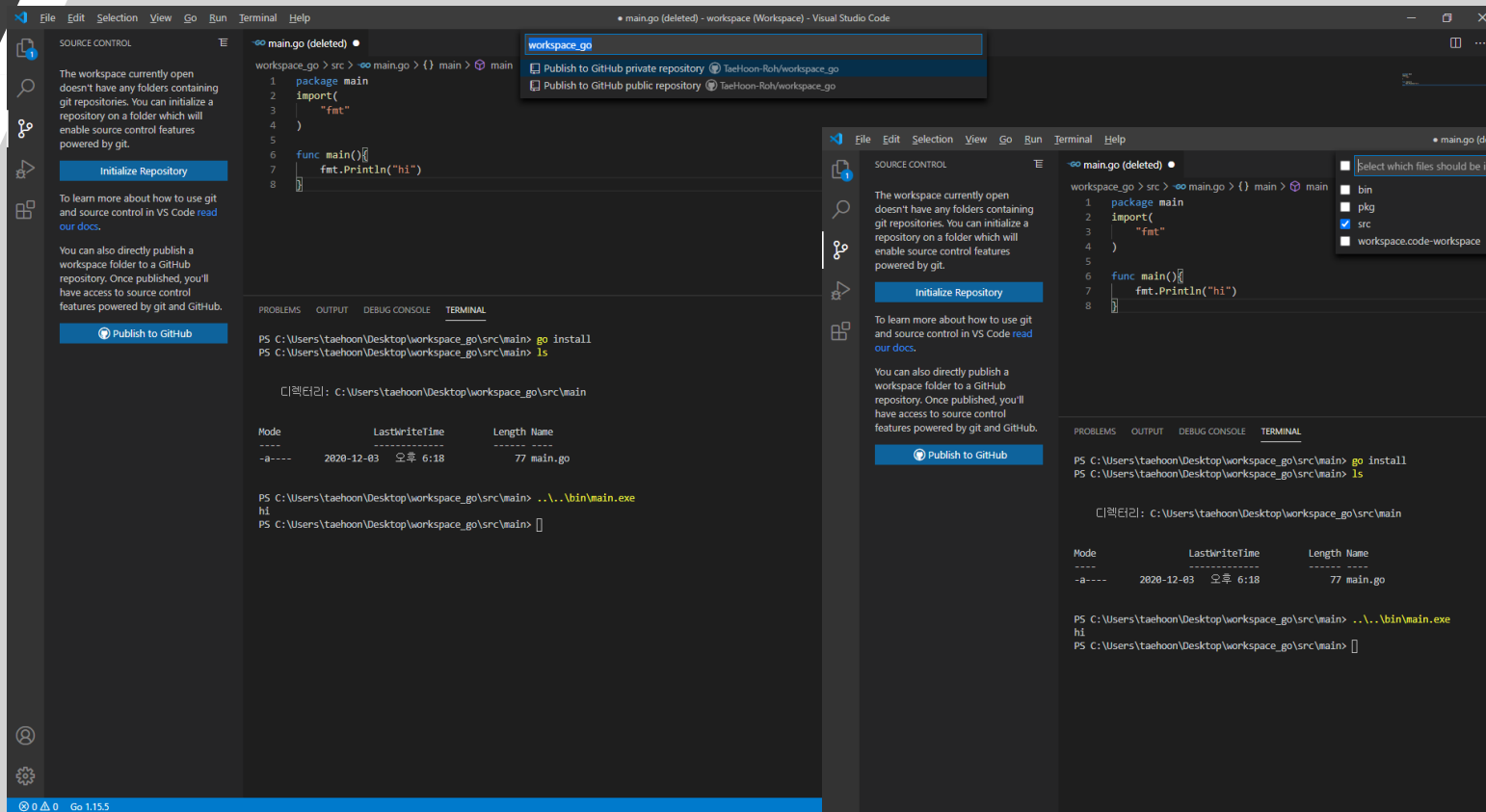
Git 설치



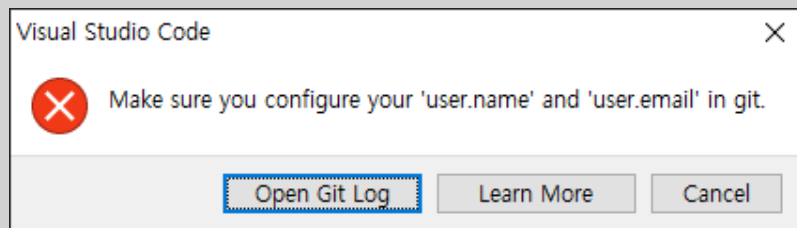
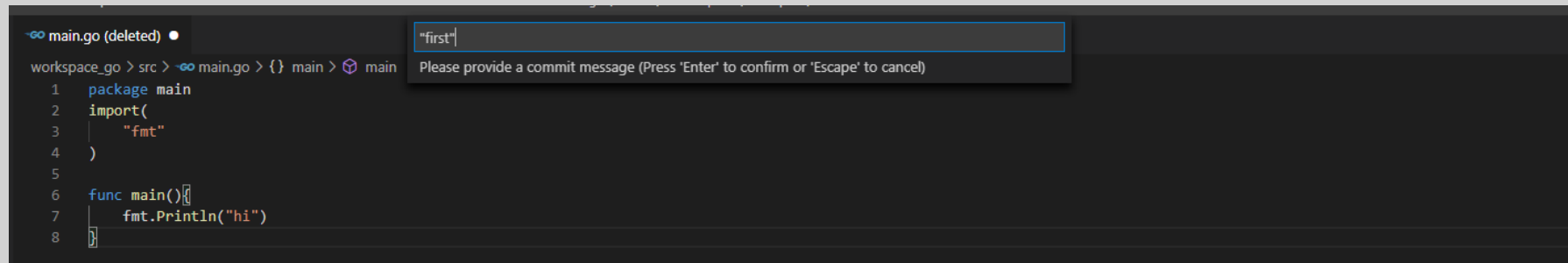
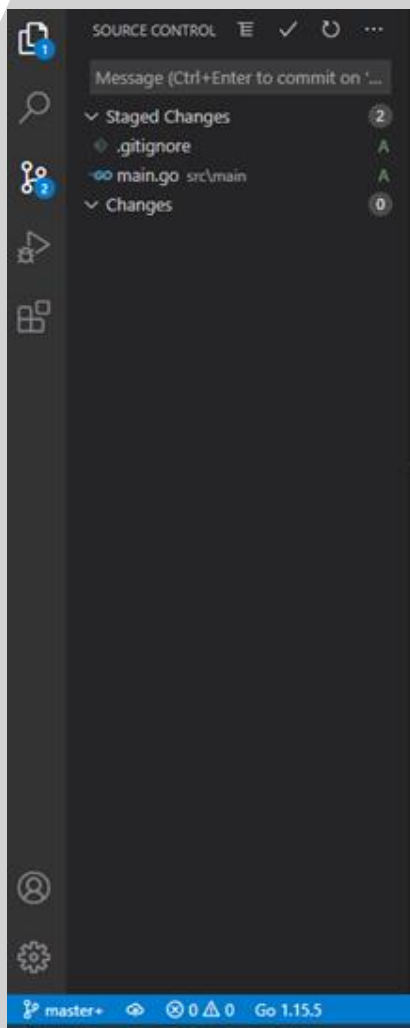
Git 설치



Git 설치



Git 설치



*** Please tell me who you are.

Run

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

to set your account's default identity.

Omit --global to set the identity only in this repository.

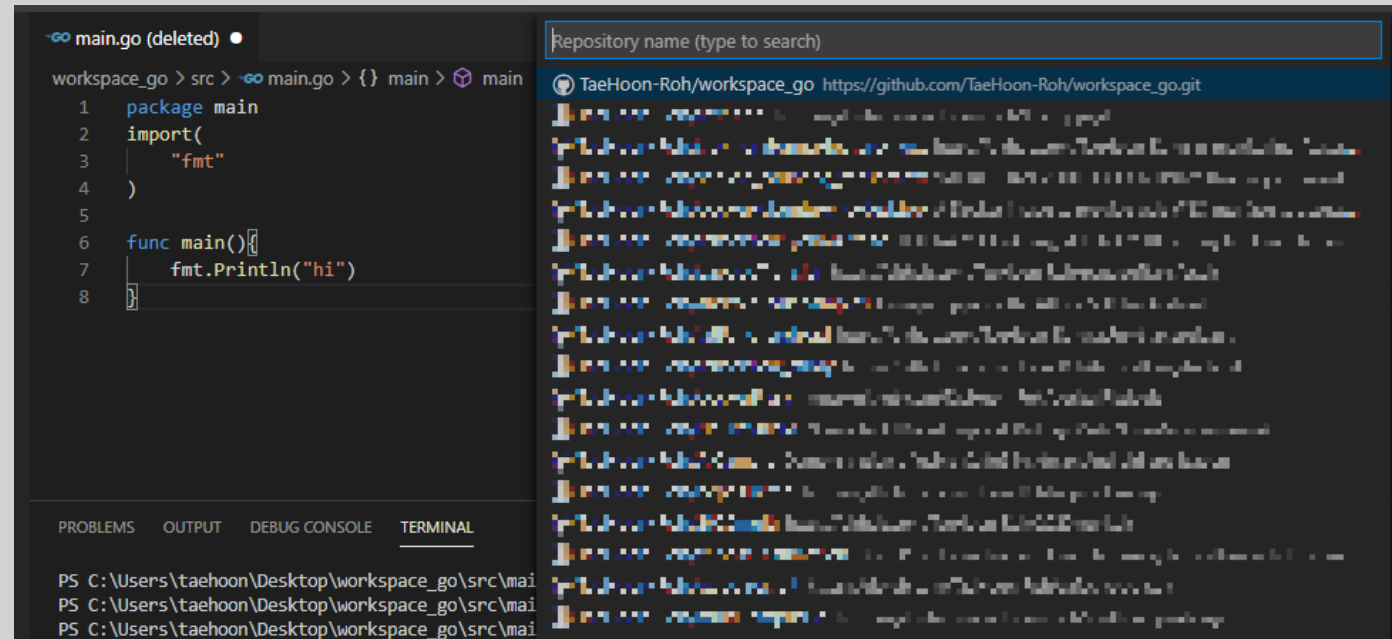
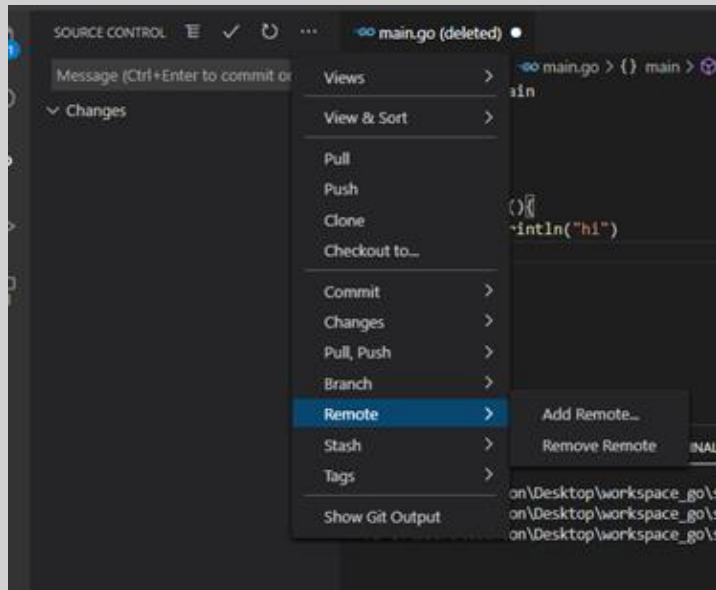
```
fatal: unable to auto-detect email address (got 'taehoon@DESKTOP-JMEGAK3.(none)')
```

```
> git config --get-all user.name
```

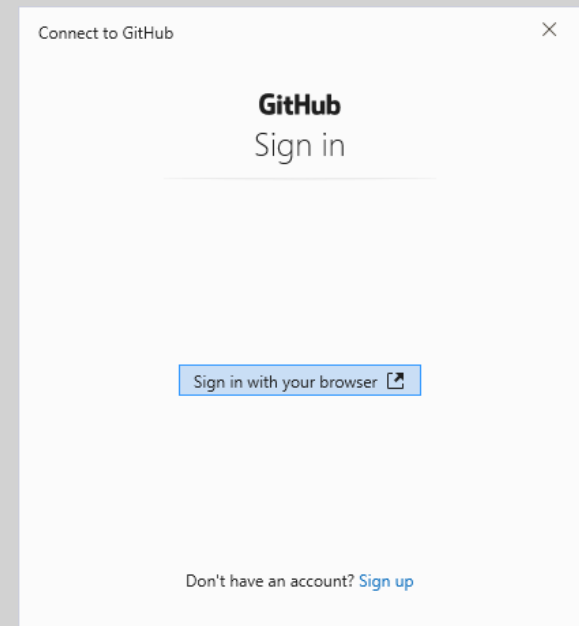
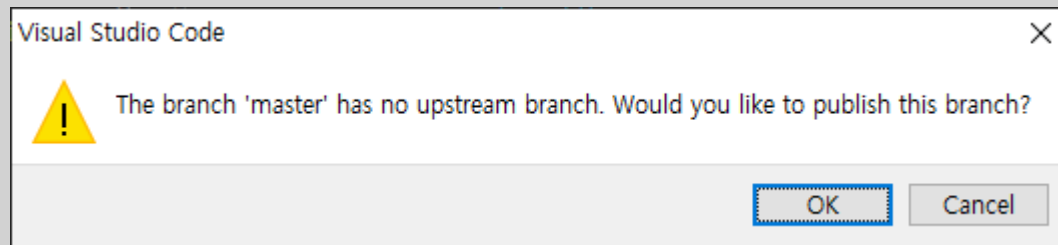
Git 설치

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\taehoon\Desktop\workspace_go\src\main> git config --global user.email "hollyth2v@gmail.com"
PS C:\Users\taehoon\Desktop\workspace_go\src\main> git config --global user.name "ggyaowang"
PS C:\Users\taehoon\Desktop\workspace_go\src\main> |
```



Git 설치



Git 설치

TaeHoon-Roh / workspace_go

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/TaeHoon-Roh/workspace_go.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# workspace_go" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/TaeHoon-Roh/workspace_go.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/TaeHoon-Roh/workspace_go.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Custom notification settings

Looking for a little more control? Now you can choose which types of activity you'd like to be notified about per repository.

Get it

TaeHoon-Roh / workspace_go

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

TaeHoon-Roh "first" bf45b25 7 minutes ago 1 commits

src/main	"first"	7 minutes ago
.gitignore	"first"	7 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

Git 설치

