TI MCU, DSP 및 Xilinx FPGA 프로그래밍 전문가 과정

Innova Lee(이상훈)
gcccompil3r@gmail.com

How to Install Git on Windows

아래와 같이 git 을 구글에서 검색한다.

git	⌨ 🎤 🔍

전체 이미지 동영상 지도 도서 더보기   설정 도구
─────

검색결과 약 211,000,000개 (0.44초)

기존 디렉토리를 **Git** 저장소로 만들기 버전관리를 하지
아니하는 기존 프로젝트를 **Git**으로 관리하고 싶은 경우
우선 프로젝트의 디렉토리로 이동한다.

git-scm.com

Git - Git 저장소 만들기
https://git-scm.com/book/ko/Git의-기초-Git-저장소-만들기

❓ 이 결과에 관한 정보   🚩 사용자 의견

Git
https://git-scm.com/ ▾ 이 페이지 번역하기

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features ...
Downloads · Documentation · Windows · About The advantages of Git ...

git --fast-version-control

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development

Downloads

🍎 **Mac OS X**	⊞ **Windows**
🐧 **Linux/Unix**	

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.18.0
Release Notes (2018-06-21)

Download 2.18.0 for Windows

GUI Clients

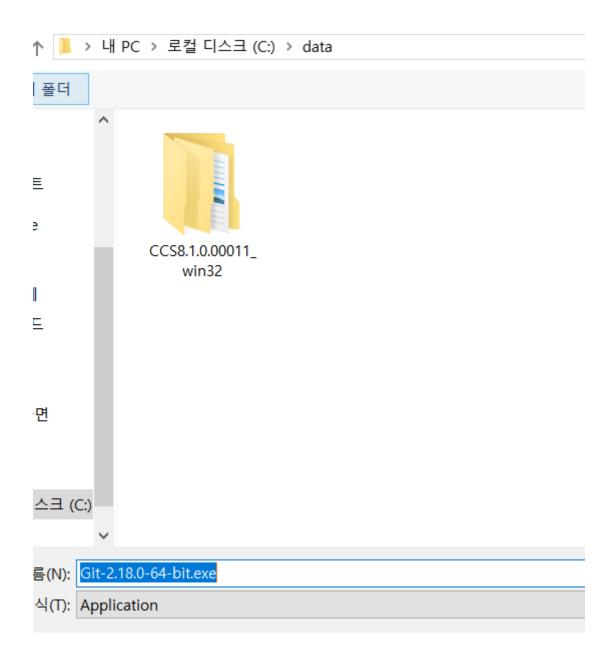Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.
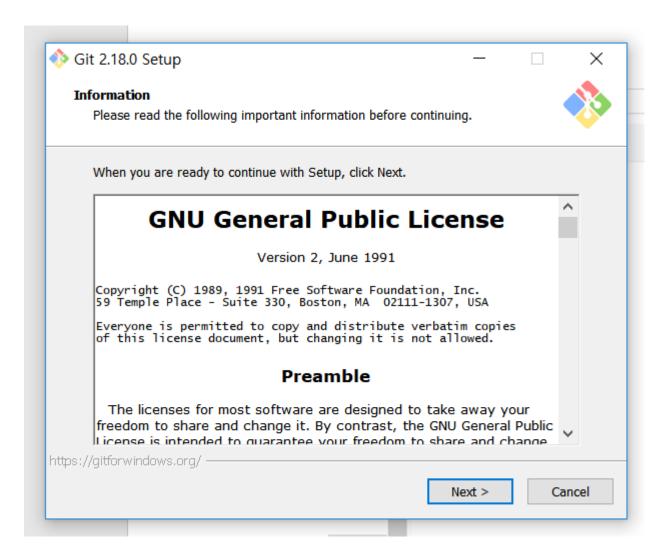
View GUI Clients →
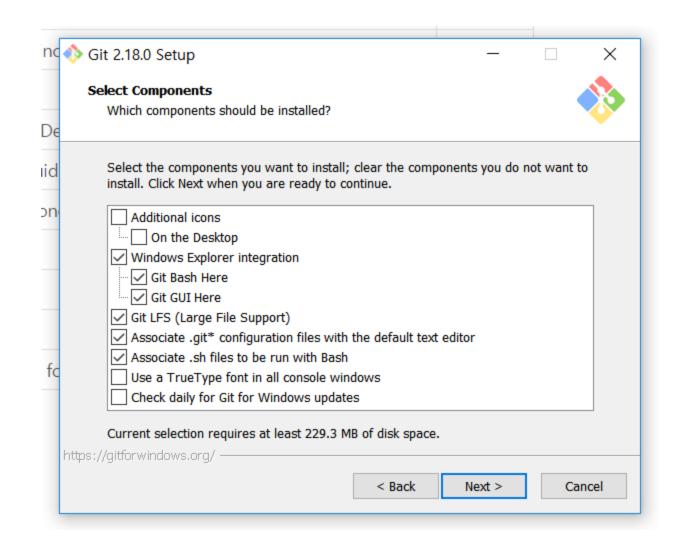
Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

내 PC > 로컬 디스크 (C:) > data

폴더

CCS8.1.0.00011_
win32

름(N): Git-2.18.0-64-bit.exe

식(T): Application

Git 2.18.0 Setup — □ ✕

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

  The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

Next >   Cancel

Git 2.18.0 Setup

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [ ] Additional icons
  - [ ] On the Desktop
- [x] Windows Explorer integration
  - [x] Git Bash Here
  - [x] Git GUI Here
- [x] Git LFS (Large File Support)
- [x] Associate .git* configuration files with the default text editor
- [x] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows
- [ ] Check daily for Git for Windows updates

Current selection requires at least 229.3 MB of disk space.

https://gitforwindows.org/

< Back   Next >   Cancel

Git 2.18.0 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor ▾

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, an it is highly recommended to switch to a modern GUI editor instead.

< Back   Next >   Cancel

Git 2.18.0 Setup

Adjusting your PATH environment
How would you like to use Git from the command line?

( ) Use Git from Git Bash only

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

( ) Use Git from the Windows Command Prompt

This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

( ) Use Git and optional Unix tools from the Windows Command Prompt

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

[< Back] [Next >] [Cancel]

Git 2.18.0 Setup

Choosing the SSH executable
Which Secure Shell client program would you like Git to use?

◉ **Use OpenSSH**

This uses ssh.exe that comes with Git. The GIT_SSH and SVN_SSH environment variables will not be modified.

○ **Use (Tortoise)Plink**

PuTTY sessions were found in your Registry. You may specify the path to an existing copy of (Tortoise)Plink.exe from the TortoiseGit/SVN/CVS or PuTTY applications. The GIT_SSH and SVN_SSH environment variables will be adjusted to point to the following executable:

C:\Program Files\PuTTY\plink.exe

https://gitforwindows.org/

[< Back] [Next >] [Cancel]

Git 2.18.0 Setup

Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

○ **Use the OpenSSL library**

    Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

    Server certificates will be validated using Windows Certificate Stores.
    This option also allows you to use your company's internal Root CA certificates
    distributed e.g. via Active Directory Domain Services.

[< Back]　[Next >]　[Cancel]

Git 2.18.0 Setup

Configuring the line ending conversions
How should Git treat line endings in text files?

○ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://gitforwindows.org/

[< Back] [Next >] [Cancel]

Git 2.18.0 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

○ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

https://gitforwindows.org/

< Back   Next >   Cancel

Git 2.18.0 Setup

Configuring extra options
Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable Git Credential Manager**

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

https://gitforwindows.org/

< Back   Install   Cancel

Git 2.18.0 Setup — □ ✕

Installing
Please wait while Setup installs Git on your computer.

Extracting files...
C:\Program Files\Git\mingw64\lib\tcl8.6\tzdata

https://gitforwindows.org/

Cancel

Git 2.18.0 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer.

Click Finish to exit Setup.

☑ Launch Git Bash

☐ View Release Notes

Finish

```
MINGW64:/c/Users/apple/AppData/Roaming/SPB_Data

apple@LAPTOP-E6V9TS3H MINGW64 ~
$
```

```
seaborn.ipynb
seaborn-data/
Searches/
SendTo@
Templates@
ti/
Untitled.ipynb
Untitled3.ipynb
Videos/
'시작 메뉴'@
'알고리즘 성능 비교.ipynb'

apple@LAPTOP-E6V9TS3H MINGW64 /c/Users/apple
$ cd ..

apple@LAPTOP-E6V9TS3H MINGW64 /c/Users
$ ls
'All Users'@   apple/   Default/   'Default User'@   desktop.ini   Public/

apple@LAPTOP-E6V9TS3H MINGW64 /c/Users
$ cd ..

apple@LAPTOP-E6V9TS3H MINGW64 /c
$ vi ~/.bashrc
```

```
cd /c/ti_workspace/git
~
~
~
~
~
~
~
~
~
~
~
~
~
~
<rs/apple/AppData/Roaming/SPB_Data/.bashrc[+] [unix] (08:59 01/01/1970)1,22 모 두
```

```
dsp_proj/                      pagefile.sys              ti_workspace/
git_proj/                      PerfLogs/                 Users/
hiberfil.sys                   'Program Files'/          vivado_dsp_proj/
Intel/                         'Program Files (x86)'/    vivado_proj/
Kings/                         ProgramData/              Windows/
MentorGraphics/                PSIM/                     Xilinx/
ocv-python-lib/                psim_workspace/

apple@LAPTOP-E6V9TS3H MINGW64 /c
$ cd ti_workspace/

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace
$ ls
RemoteSystemsTempFiles/

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace
$ mkdir git

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace
$ ls
git/   RemoteSystemsTempFiles/

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace
$ |
```