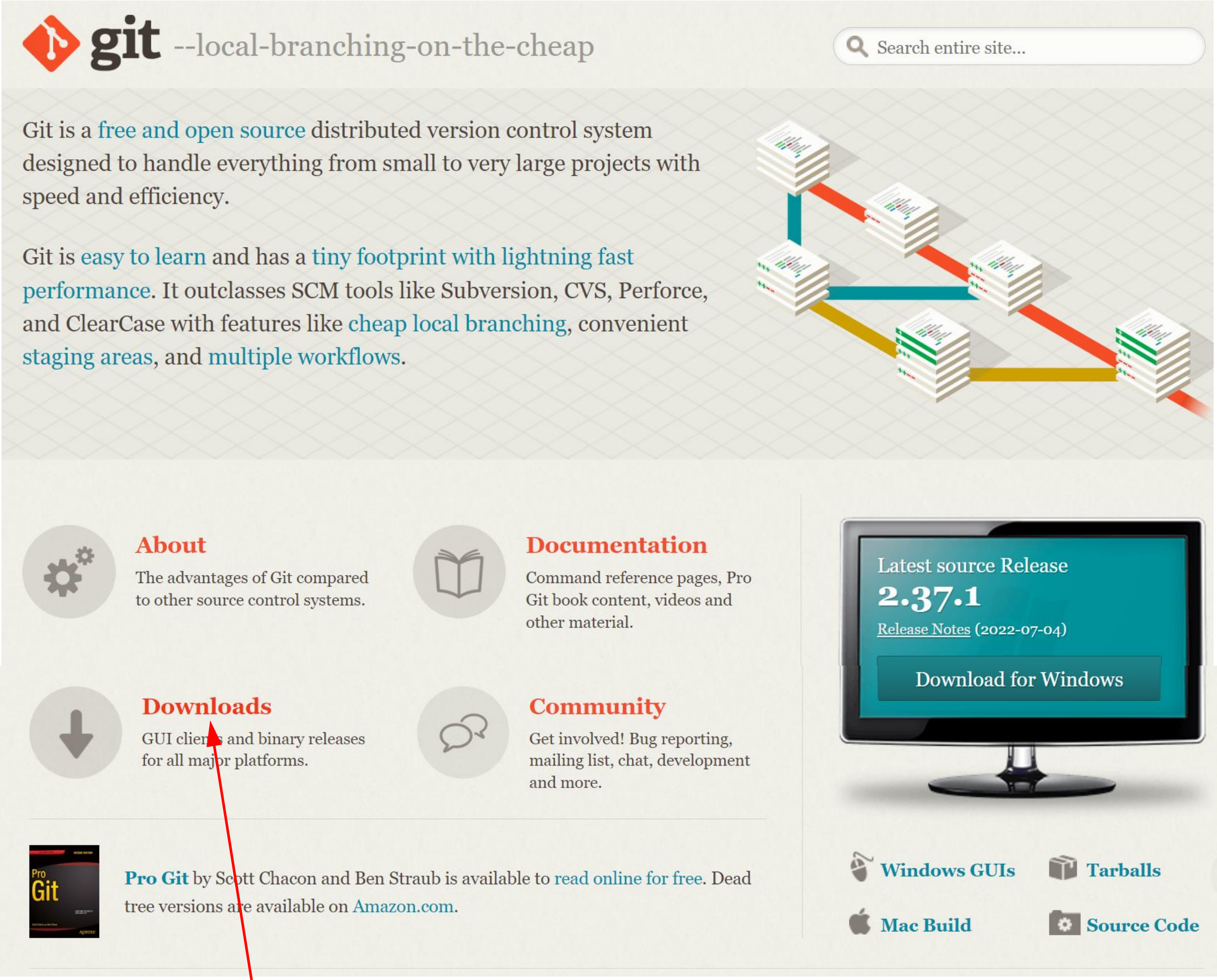
① https://git-scm.com/ にアクセス



git --local-branching-on-the-cheap

Search entire site...

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.

**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 and more.

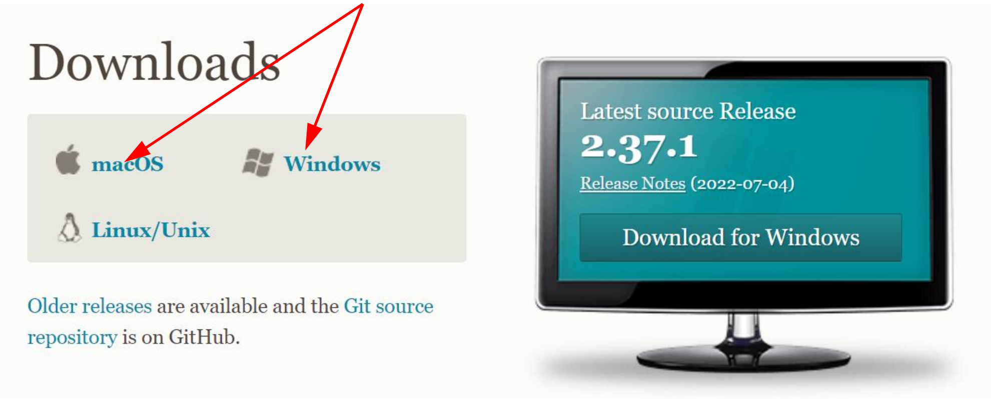
Pro Git by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Latest source Release
2.37.1
Release Notes (2022-07-04)
Download for Windows

Windows GUIs
Tarballs
Mac Build
Source Code

②クリック

③選択してクリック



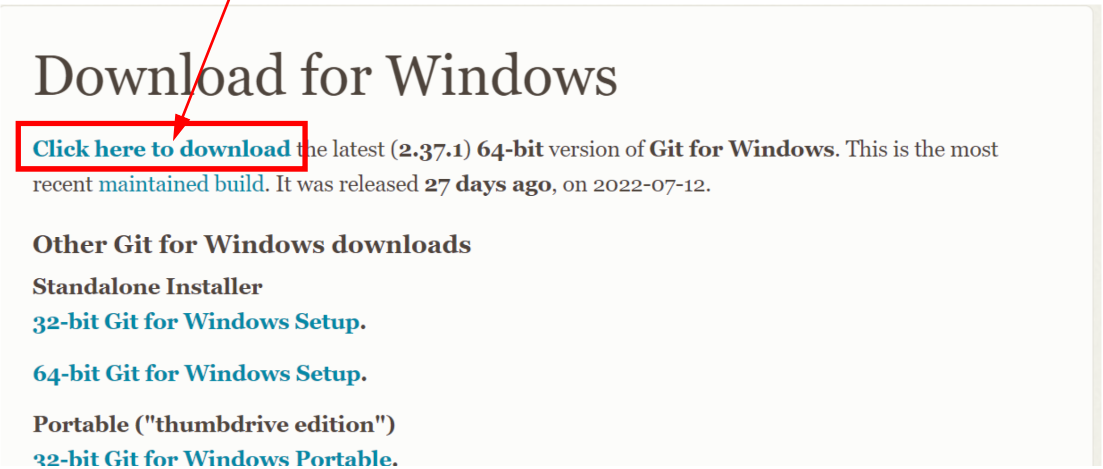
Downloads

macOS
Windows
Linux/Unix

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

Latest source Release
2.37.1
Release Notes (2022-07-04)
Download for Windows

④クリック



Download for Windows

Click here to download the latest (2.37.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 27 days ago, on 2022-07-12.

**Other Git for Windows downloads**

**Standalone Installer**

32-bit Git for Windows Setup.

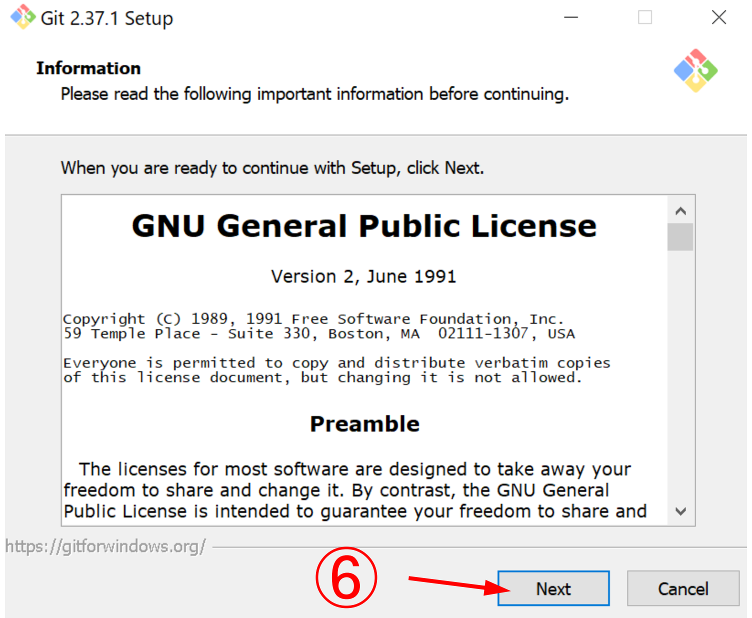
64-bit Git for Windows Setup.

**Portable ("thumbdrive edition")**

32-bit Git for Windows Portable.

⑤ダウンロードした実行ファイルクリック

Git-2.37.1-64-bit.exe



Git 2.37.1 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

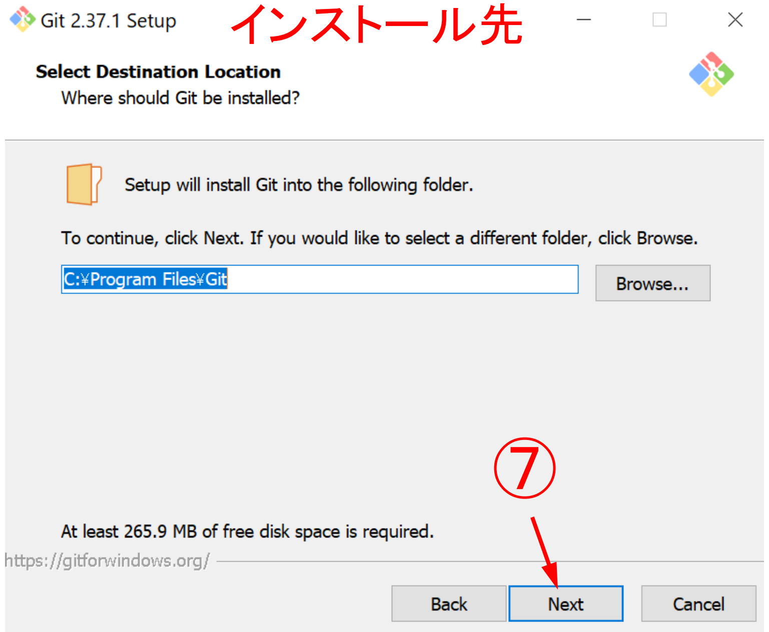
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

**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

https://gitforwindows.org/

⑥



Git 2.37.1 Setup

インストール先

**Select Destination Location**

Where should Git be installed?

Setup will install Git into the following folder.

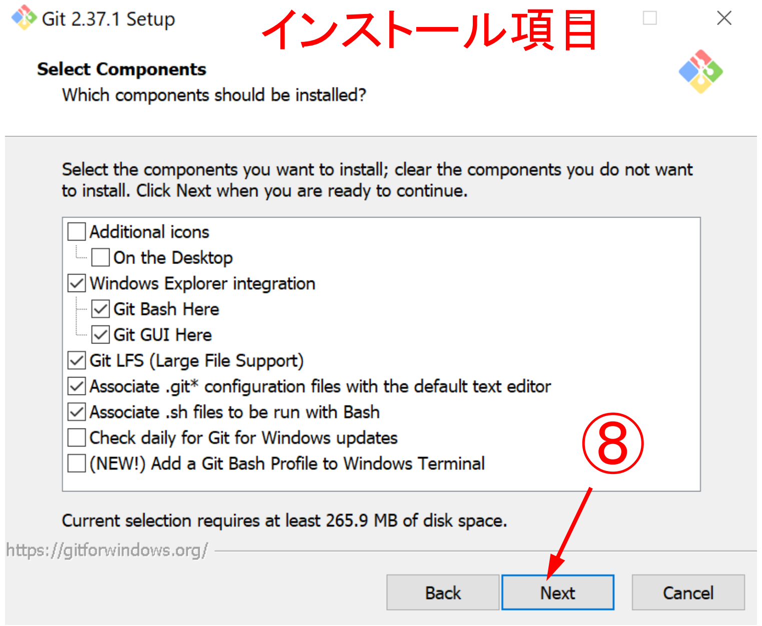
To continue, click Next. If you would like to select a different folder, click Browse.

C:¥Program Files¥Git

At least 265.9 MB of free disk space is required.

https://gitforwindows.org/

⑦



Git 2.37.1 Setup

インストール項目

**Select Components**

Which components should be installed?

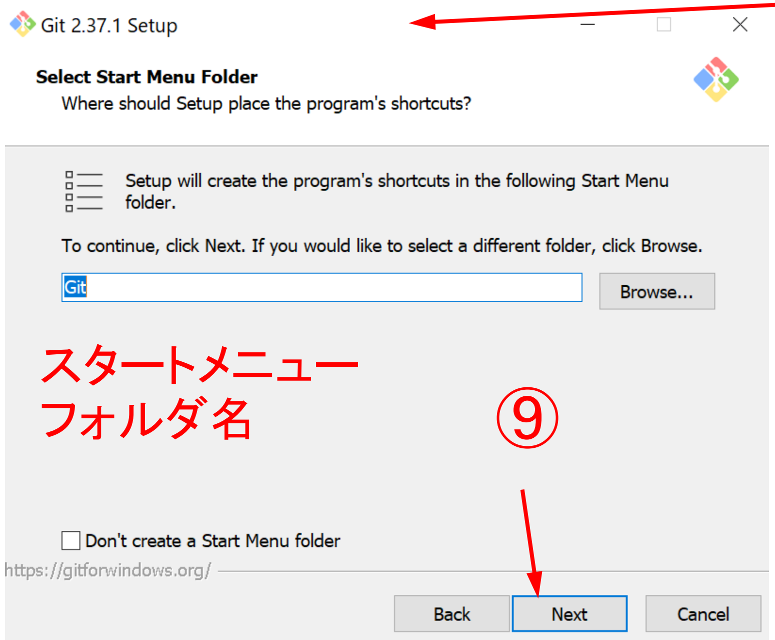
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- □ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 265.9 MB of disk space.

https://gitforwindows.org/

⑧



Git 2.37.1 Setup

**Select Start Menu Folder**

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

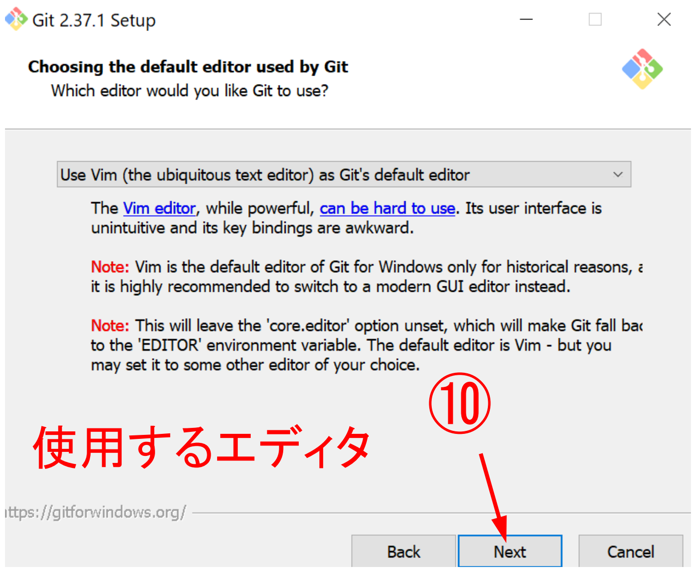
Git

スタートメニュー
フォルダ名

⑨

□ Don't create a Start Menu folder

https://gitforwindows.org/



Git 2.37.1 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.

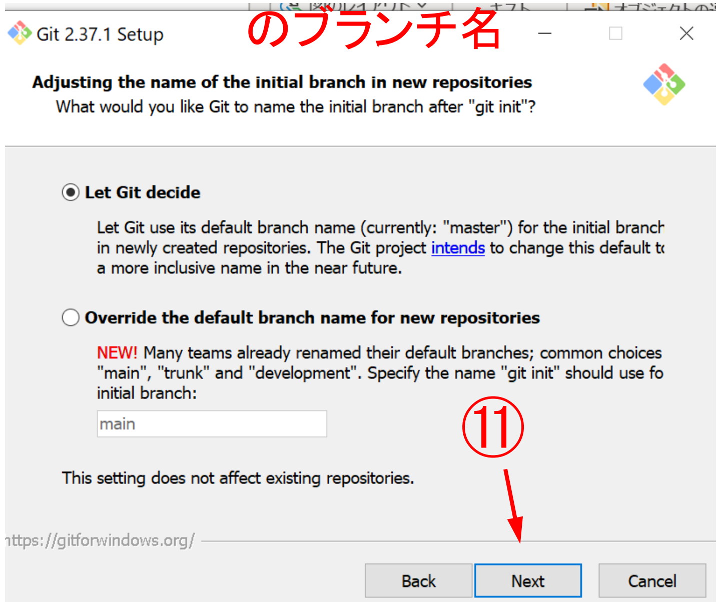
Note: Vim is the default editor of Git for Windows only for historical reasons, a
it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall bac
to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

使用するエディタ

⑩

https://gitforwindows.org/



リポジトリ作成時
のブランチ名

Git 2.37.1 Setup

**Adjusting the name of the initial branch in new repositories**

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch
in newly created repositories. The Git project intends to change this default t
a more inclusive name in the near future.

○ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices "main", "trunk" and "development". Specify the name "git init" should use fo
initial branch:

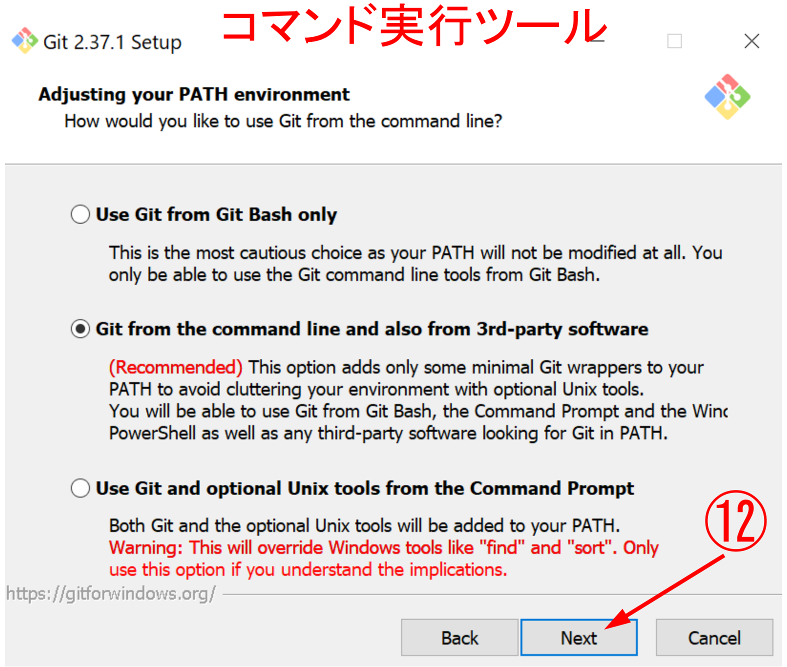
main

This setting does not affect existing repositories.

⑪

https://gitforwindows.org/

コマンド実行ツール



Git 2.37.1 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 most cautious choice as your PATH will not be modified at all. You only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Winc PowerShell as well as any third-party software looking for Git in PATH.

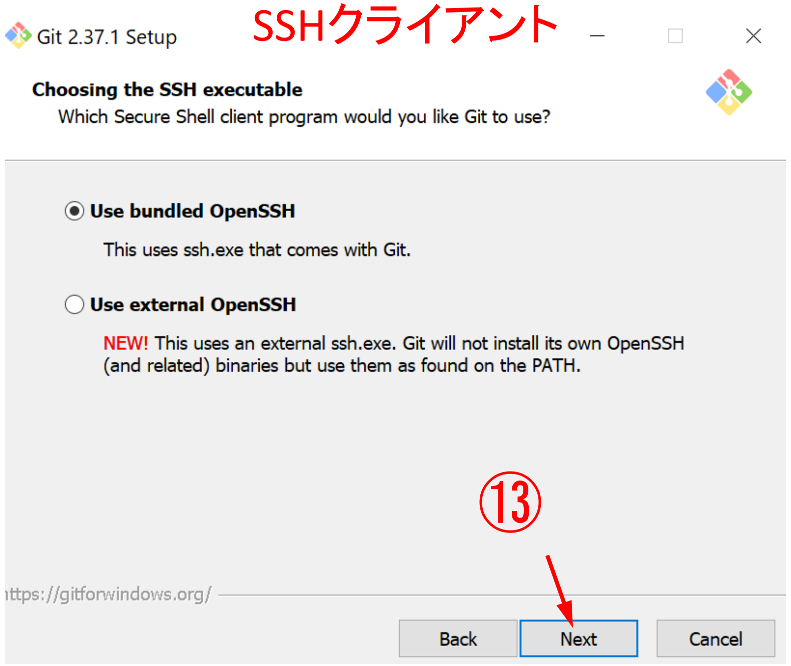
○ **Use Git and optional Unix tools from the Command Prompt**

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/

⑫

SSHクライアント



Git 2.37.1 Setup

**Choosing the SSH executable**

Which Secure Shell client program would you like Git to use?

● **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

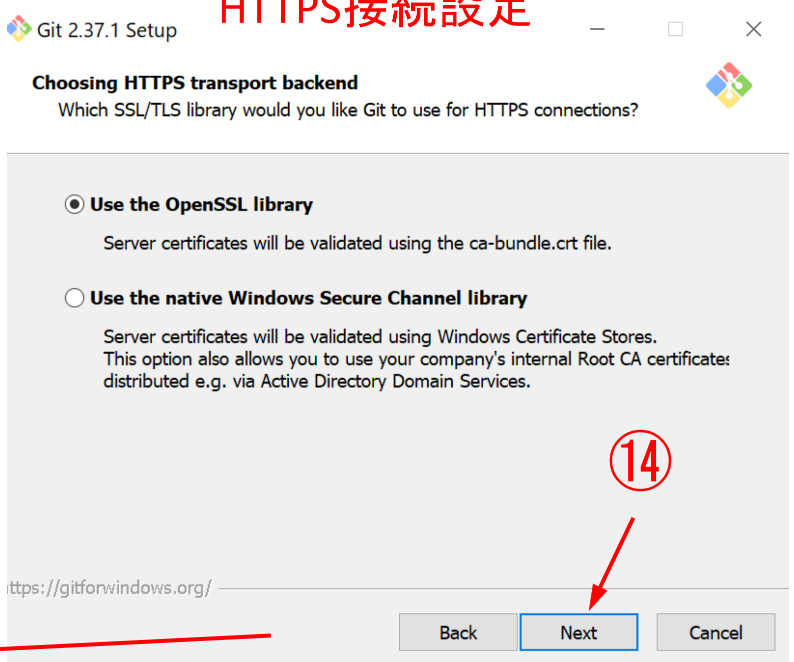
○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

ttps://gitforwindows.org/

⑬

HTTPS接続設定



Git 2.37.1 Setup

**Choosing HTTPS transport backend**

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

● **Use the OpenSSL library**

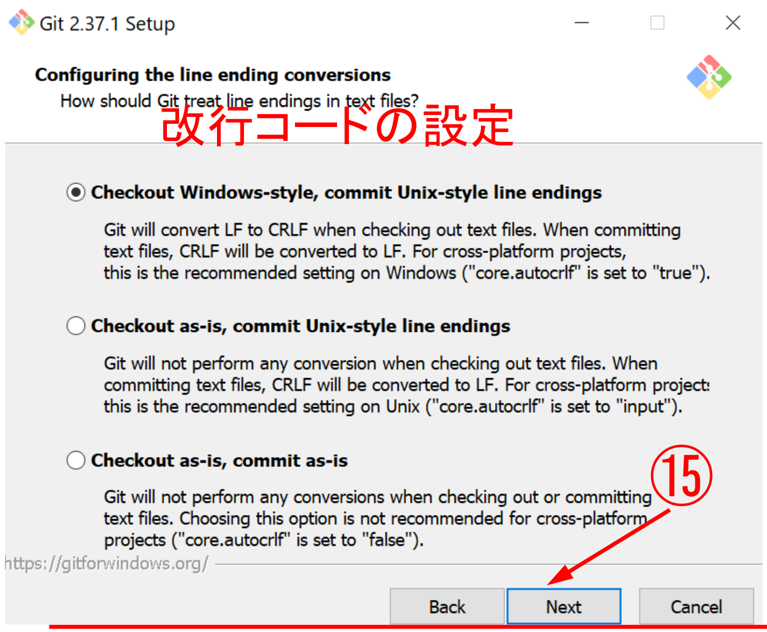
Server certificates will be validated using the ca-bundle.crt file.

○ **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.

ttps://gitforwindows.org/

⑭



Git 2.37.1 Setup

**Configuring the line ending conversions**

How should Git treat line endings in text files?

改行コードの設定

● **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**

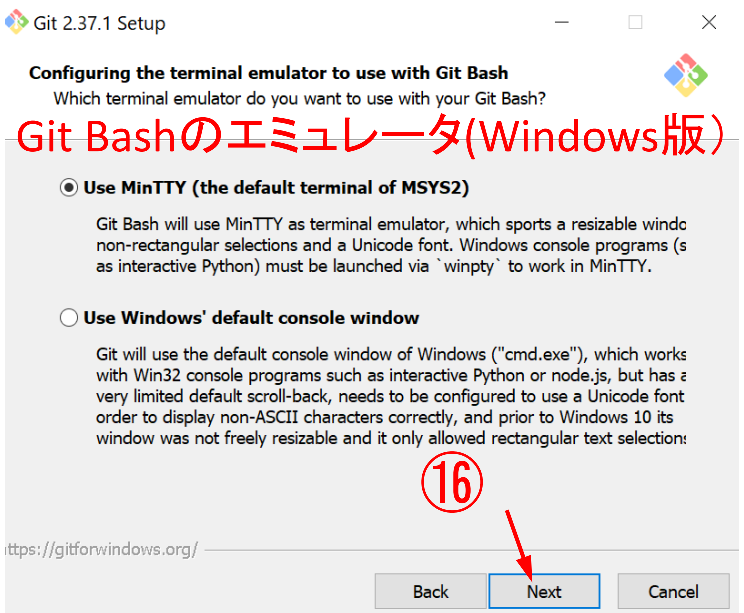
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform project: this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **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/

⑮



Git 2.37.1 Setup

**Configuring the terminal emulator to use with Git Bash**

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

Git Bashのエミュレータ(Windows版)

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

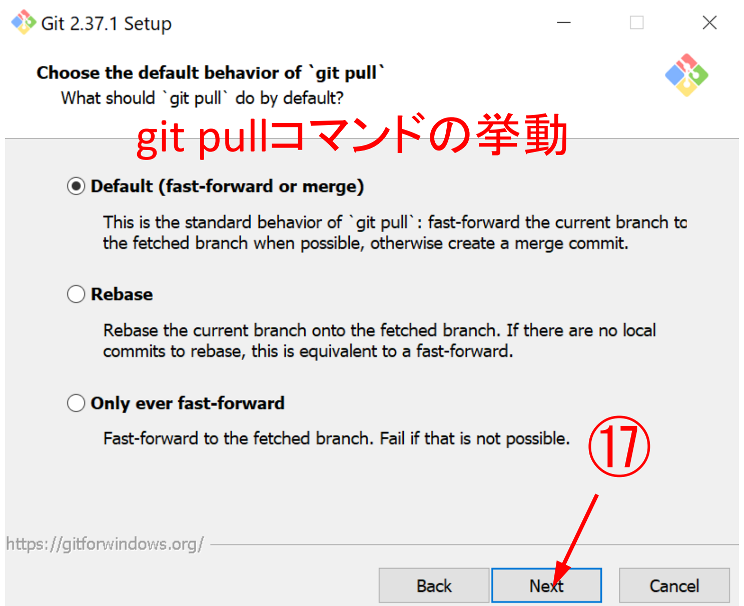
Git Bash will use MinTTY as terminal emulator, which sports a resizable windc non-rectangular selections and a Unicode font. Windows console programs (s as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **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 order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selection:

ttps://gitforwindows.org/

⑯



Git 2.37.1 Setup

**Choose the default behavior of `git pull`**

What should `git pull` do by default?

git pullコマンドの挙動

● **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: fast-forward the current branch tc the fetched branch when possible, otherwise create a merge commit.

○ **Rebase**

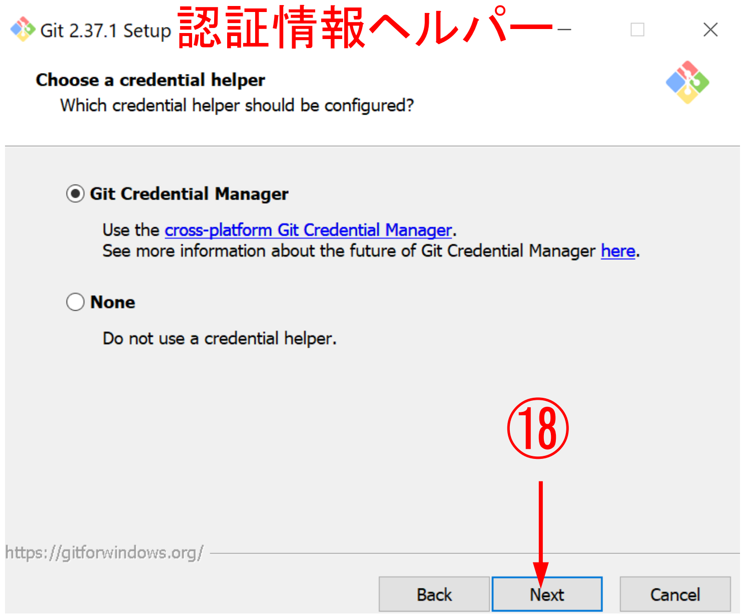
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

○ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.

https://gitforwindows.org/

⑰



Git 2.37.1 Setup 認証情報ヘルパー

**Choose a credential helper**

Which credential helper should be configured?

● **Git Credential Manager**

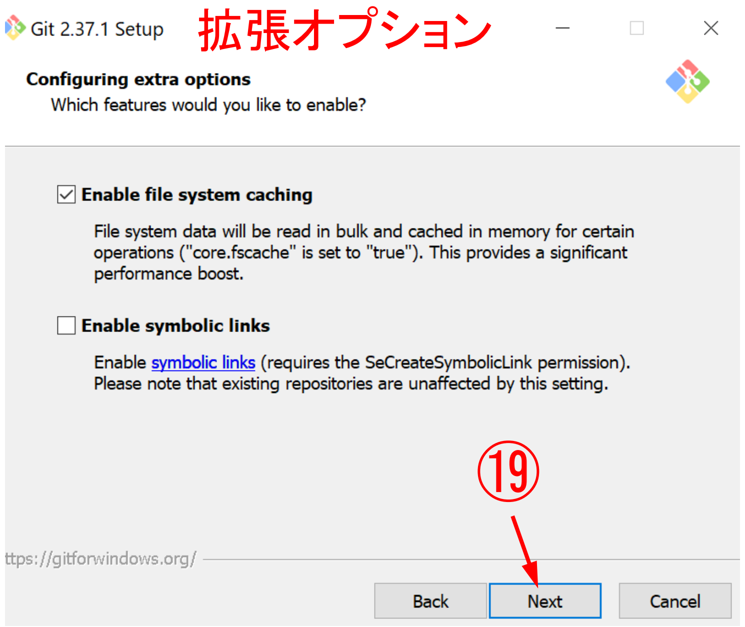
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ **None**

Do not use a credential helper.

https://gitforwindows.org/

⑱



Git 2.37.1 Setup 拡張オプション

**Configuring extra options**

Which features would you like to enable?

☑ **Enable file system caching**

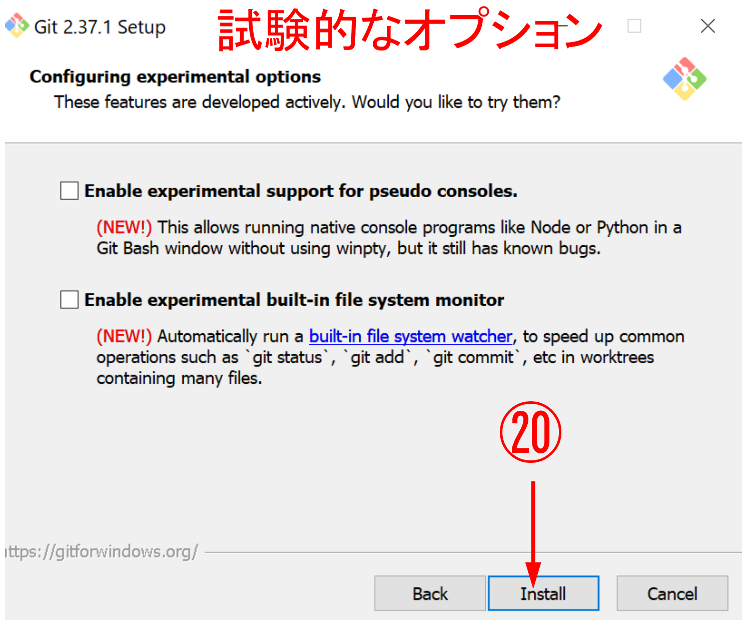
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

□ **Enable symbolic links**

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

ttps://gitforwindows.org/

⑲



Git 2.37.1 Setup 試験的なオプション

**Configuring experimental options**

These features are developed actively. Would you like to try them?

□ **Enable experimental support for pseudo consoles.**

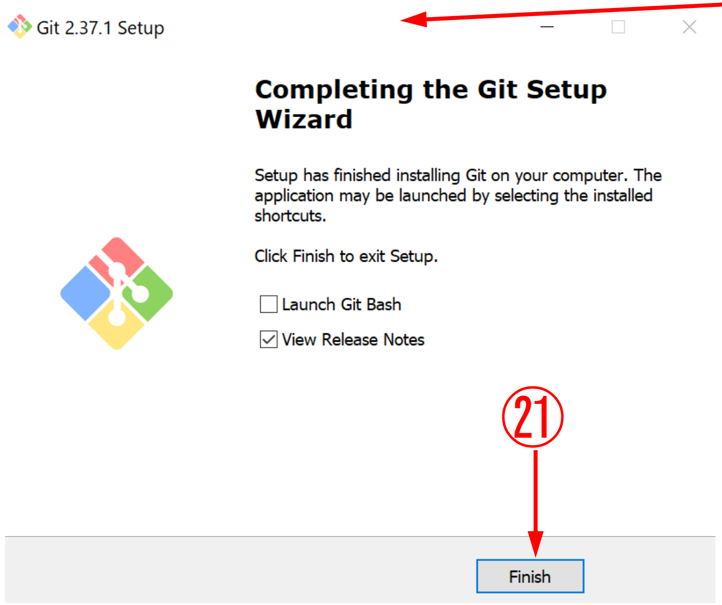
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

□ **Enable experimental built-in file system monitor**

(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.

ttps://gitforwindows.org/

⑳



Git 2.37.1 Setup

**Completing the Git Setup Wizard**

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

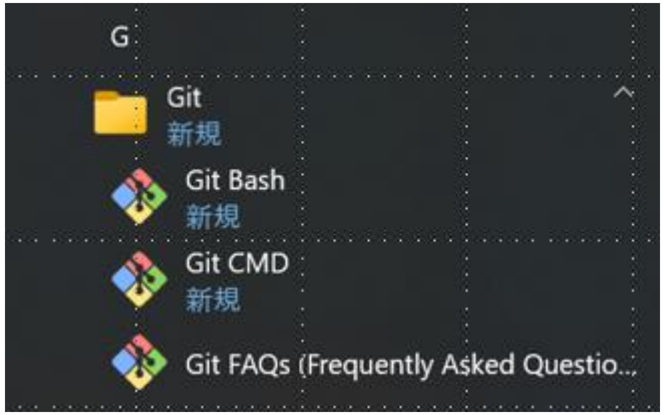
Click Finish to exit Setup.

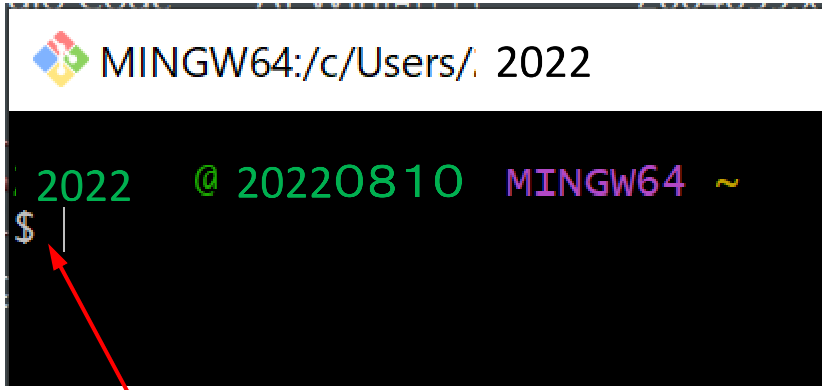
□ Launch Git Bash

☑ View Release Notes

㉑

㉒Windowsスタートメニューで
Git Bashをクリック
WindowsでもmacOsでも
同じコマンドで実行できる

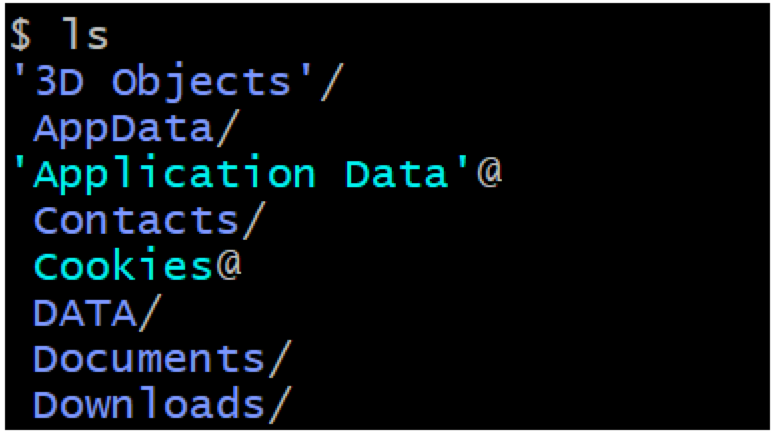




```
2022   @ 20220810  MINGW64 ~
$ |
```

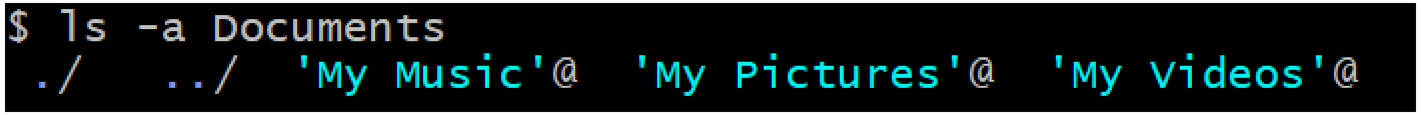
＄　プロンプト

## コマンド　ls　ディレクトリの内容表示



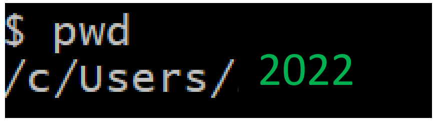
```
$ ls
'3D Objects'/
 AppData/
'Application Data'@
 Contacts/
 Cookies@
 DATA/
 Documents/
 Downloads/
```

## コマンド　ls　ーa　　オプション「.」で始まるディレクトリを表示



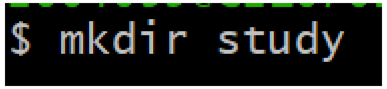
```
$ ls -a Documents
 ./    ../   'My Music'@   'My Pictures'@   'My Videos'@
```

## コマンド　pwd　絶対パスを表示



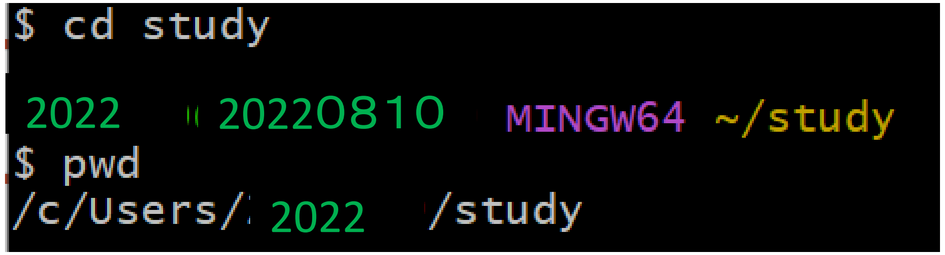
```
$ pwd
/c/Users/ 2022
```

## コマンド mkdir ディレクトリを作成



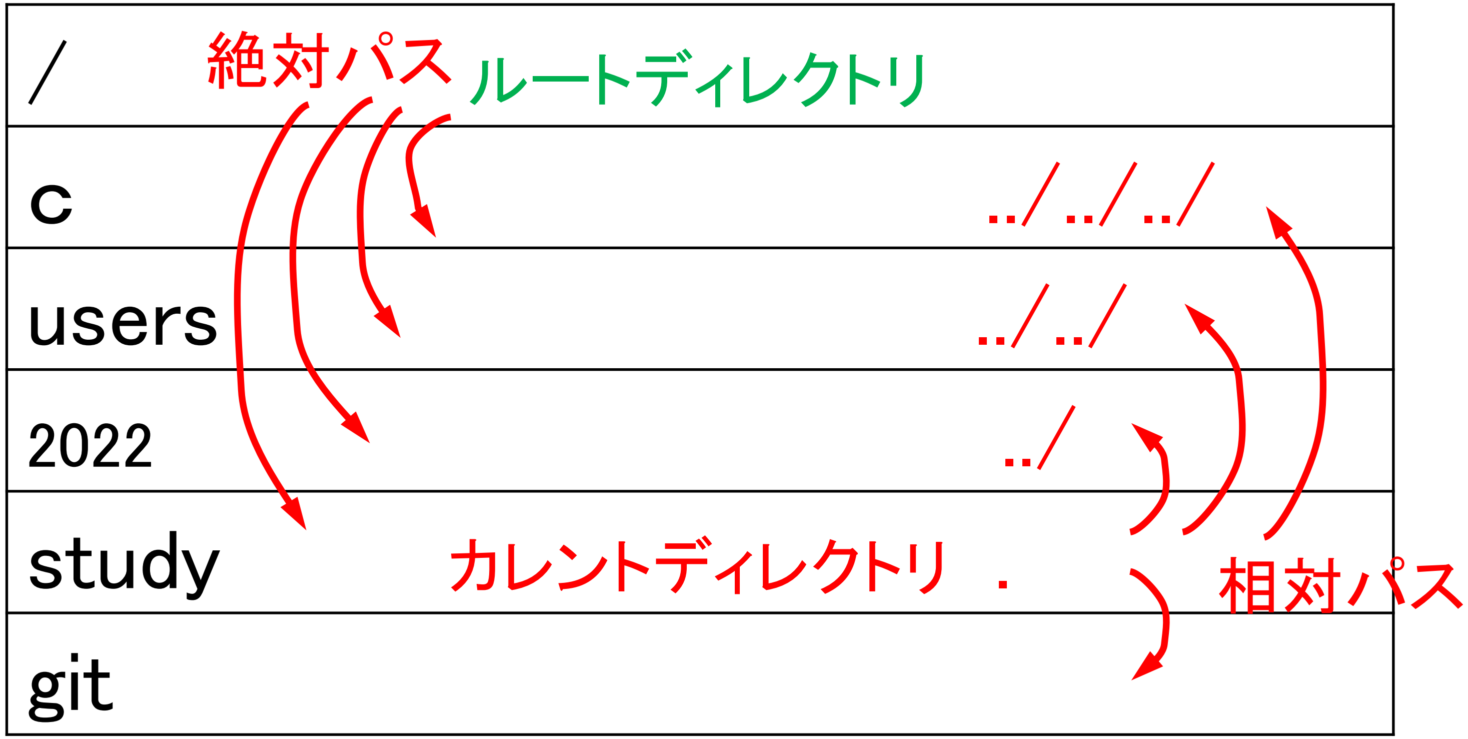
```
$ mkdir study
```

## コマンド cd ディレクトリを移動

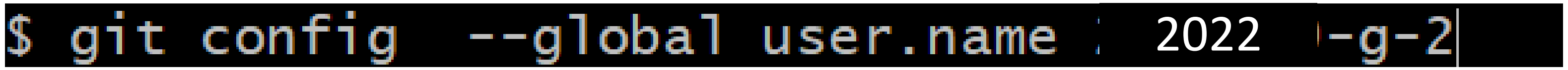


```
$ cd study

 2022     20220810    MINGW64 ~/study
$ pwd
/c/Users/ 2022    /study
```

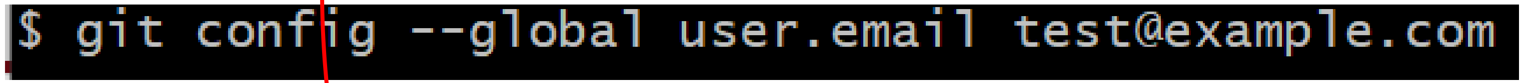


## ㉓ユーザー名設定

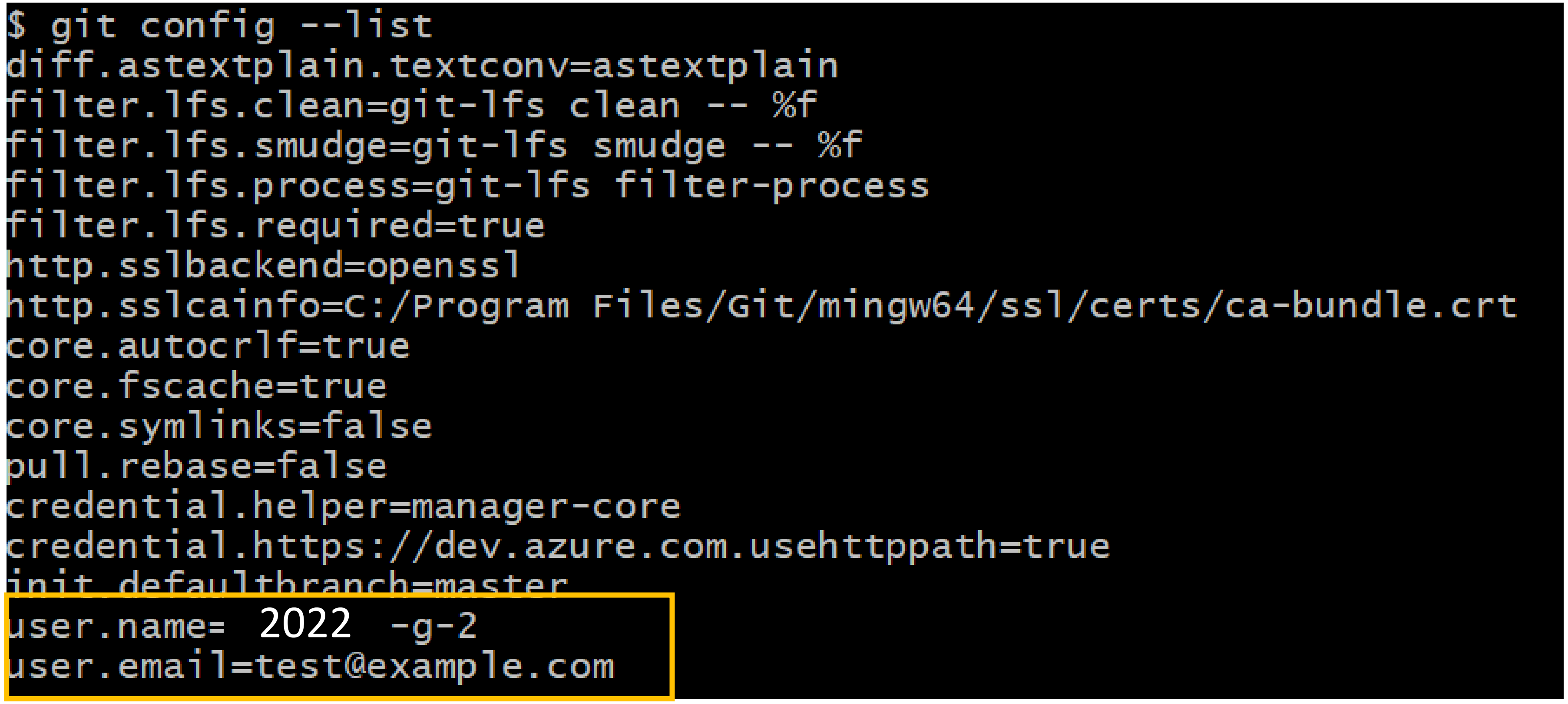


```
$ git config  --global user.name [redacted] 2022 -g-2
```

## ㉔メールアドレス設定

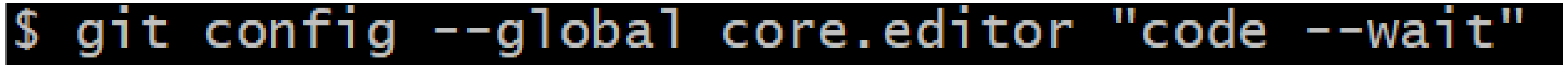


```
$ git config --global user.email test@example.com
```

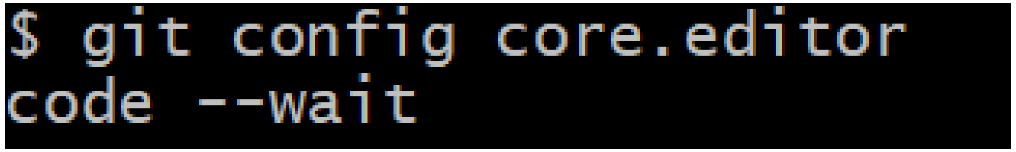


```
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=[redacted] 2022 -g-2
user.email=test@example.com
```

Visual Studio Codeをエディターに用います。
自動で切り替えるように設定しておきます。



```
$ git config --global core.editor "code --wait"
```

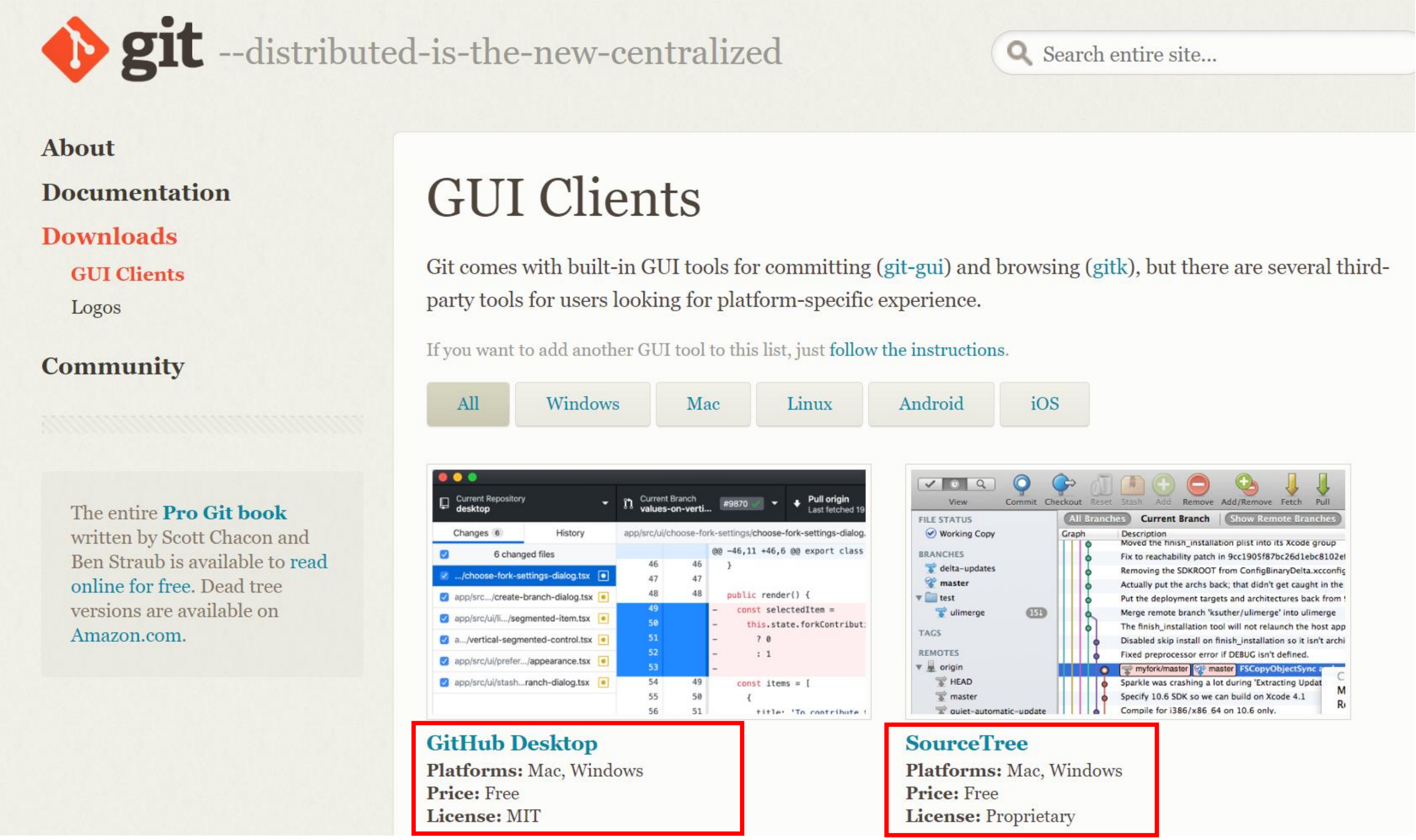


```
$ git config core.editor
code --wait
```

# ｸﾞﾗﾌｨｶﾙ画面　GUI(Graphical User Interface)を使用したい場合

プロンプトで操作する場合は、CUI(Character User Interface)

https://git-scm.com/downloads/guis　のGUI紹介ページににアクセス



git --distributed-is-the-new-centralized

Search entire site...

- About
- Documentation
- Downloads
  - GUI Clients
  - Logos
- Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

## GUI Clients

Git comes with built-in GUI tools for committing (git-gui) and browsing (gitk), but there are several third-party tools for users looking for platform-specific experience.

If you want to add another GUI tool to this list, just follow the instructions.

All | Windows | Mac | Linux | Android | iOS

**GitHub Desktop**
**Platforms:** Mac, Windows
**Price:** Free
**License:** MIT

**SourceTree**
**Platforms:** Mac, Windows
**Price:** Free
**License:** Proprietary