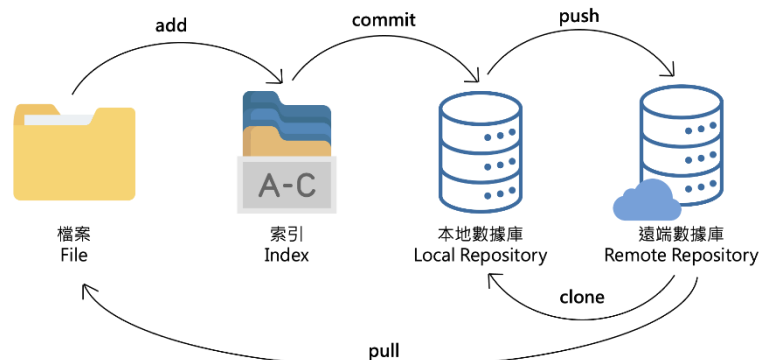


Git - the simple guide

by Roger Dudler

1. Git 簡介：Git 是一個版本控制系統，而 GitHub 是使用 Git 來管理檔案的平台。



圖片來源：<https://ithelp.ithome.com.tw/articles/10266796>

- (1) 本地儲存庫(Local Repository)：利用 `git clone` 指令可將 github、gitlab 網站等(遠端儲存庫，Remote Repository)的 Repository 複製一份到本地端進行操作。操作完成後，透過 `git add` 將工作目錄中的變更添加到暫存區域。告訴 Git 想在下一次提交中包含對特定文件的更新。
- (2) 經由 `git commit` 後才會真正將記錄提交；可經由 `git status` 指令檢視工作目錄與暫存區的狀態。最後再經由 `git push` 指令將本機端儲存庫的相關檔案，上傳到遠端儲存庫。相關指令與步驟可參考下圖。

INSTALLATION & GUIs

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows
<https://windows.github.com>

GitHub for Mac
<https://mac.github.com>

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms
<http://git-scm.com>

SETUP

Configuring user information used across all local repositories

```
git config --global user.name "[firstname lastname]"
set a name that is identifiable for credit when review version history

git config --global user.email "[valid-email]"
set an email address that will be associated with each history marker

git config --global color.ui auto
set automatic command line coloring for Git for easy reviewing
```

SETUP & INIT

Configuring user information, initializing and cloning repositories

```
git init
initialize an existing directory as a Git repository

git clone [url]
retrieve an entire repository from a hosted location via URL
```

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

```
git status
show modified files in working directory, staged for your next commit

git add [file]
add a file as it looks now to your next commit (stage)

git reset [file]
unstage a file while retaining the changes in working directory

git diff
diff of what is changed but not staged

git diff --staged
diff of what is staged but not yet committed

git commit -m "[descriptive message]"
commit your staged content as a new commit snapshot
```

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

```
git branch
list your branches. a* will appear next to the currently active branch

git branch [branch-name]
create a new branch at the current commit

git checkout
switch to another branch and check it out into your working directory

git merge [branch]
merge the specified branch's history into the current one

git log
show all commits in the current branch's history
```

<http://rogerdudler.github.io/git-guide/>
<https://w3c.hexschool.com/git/cfdbd310>

INSPECT & COMPARE

Examining logs, diffs and object information

git log
show the commit history for the currently active branch
git log branchB...branchA
show the commits on branchA that are not on branchB
git log --follow [file]
show the commits that changed file, even across renames
git diff branchB...branchA
show the diff of what is in branchA that is not in branchB
git show [SHA]
show any object in Git in human-readable format

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

git remote add [alias] [url]
add a git URL as an alias
git fetch [alias]
fetch down all the branches from that Git remote
git merge [alias]/[branch]
merge a remote branch into your current branch to bring it up to date
git push [alias] [branch]
Transmit local branch commits to the remote repository branch
git pull
fetch and merge any commits from the tracking remote branch

TRACKING PATH CHANGES

Versioning file removes and path changes

git rm [file]
delete the file from project and stage the removal for commit
git mv [existing-path] [new-path]
change an existing file path and stage the move
git log --stat -M
show all commit logs with indication of any paths that moved

REWRITE HISTORY

Rewriting branches, updating commits and clearing history

git rebase [branch]
apply any commits of current branch ahead of specified one
git reset --hard [commit]
clear staging area, rewrite working tree from specified commit

IGNORING PATTERNS

Preventing unintentional staging or committing of files

logs/ *.notes pattern*/
Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.
git config --global core.excludesfile [file]
system wide ignore pattern for all local repositories

TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

git stash
Save modified and staged changes
git stash list
list stack-order of stashed file changes
git stash pop
write working from top of stash stack
git stash drop
discard the changes from top of stash stack

圖片來源：<https://zhuanlan.zhihu.com/p/428802002>

2. 下載：[官網](#)

- (1) Git for Windows: <https://git-scm.com/download/win>
- (2) Git for Mac: <https://git-scm.com/download/mac>
- (3) Git for Linux: <https://git-scm.com/download/linux>

Downloads

macOS

Windows

Linux/Unix

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

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 →](#)



3. 安裝與設定

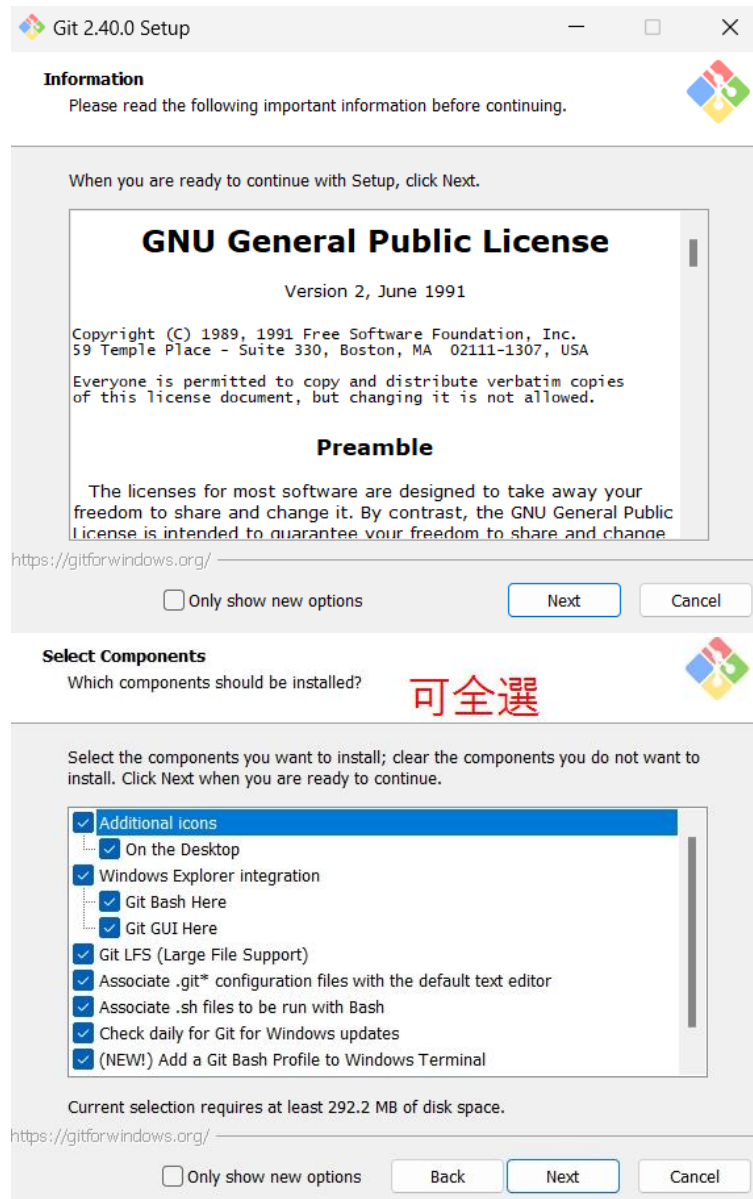
以 Windows 安裝為例，下載 64 位元版本

<http://rogerdudler.github.io/git-guide/>
<https://w3c.hexschool.com/git/cfdbd310>

Download for Windows

[Click here to download](#) the latest (2.40.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 12 days ago, on 2023-03-14.

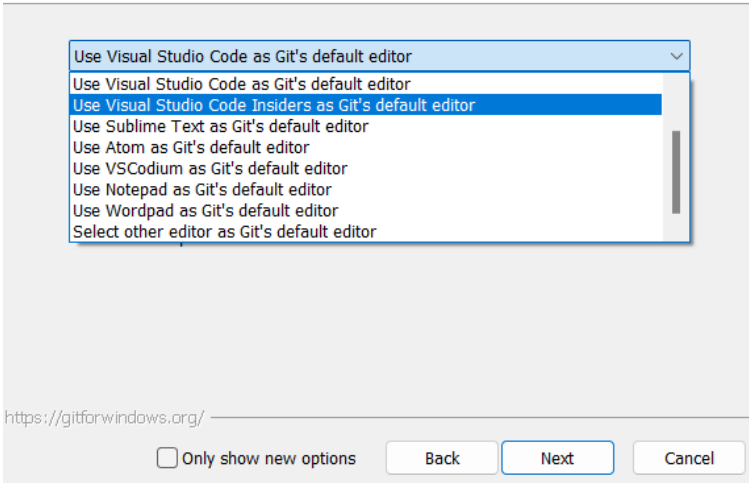

安裝教學可參考官網，<https://git-scm.com/book/zh-tw/v2/>開始-Git-安裝教學
以下拆解詳細步驟：



下拉選取習慣的編輯器，這裡依據個人習慣進行選擇

<http://rogerdudler.github.io/git-guide/>
<https://w3c.hexschool.com/git/cfdbd310>

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

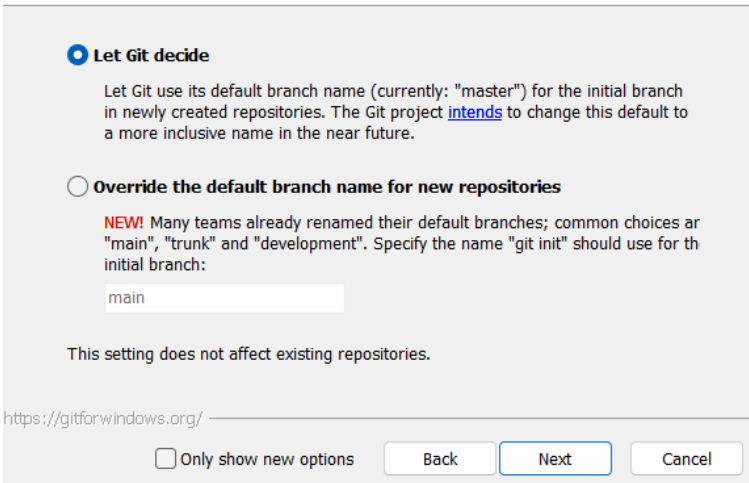



<https://gitforwindows.org/>

☐ Only show new options Back Next Cancel

是否要調整初始的分支(預設為 master)，如不更改就選擇由 Git 決定

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 to 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 are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

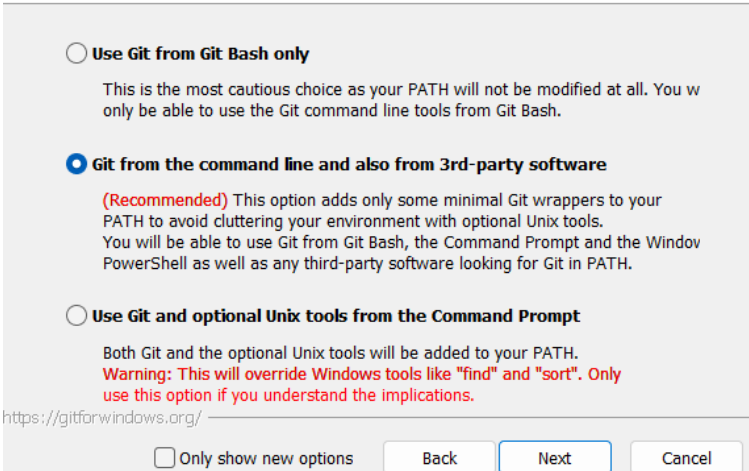

This setting does not affect existing repositories.

<https://gitforwindows.org/>

☐ Only show new options Back Next Cancel

設定預設的 Git command line 環境，使用建議的即可

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 will 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 Windows 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/>

☐ Only show new options Back Next Cancel

設定使用的 SSH 連線程式，如果沒有其他 SSH 程式，使用預設即可

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.

<https://gitforwindows.org/>

☐ Only show new options Back Next Cancel

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.

<https://gitforwindows.org/>

☐ Only show new options Back Next Cancel

設定 Git 指令結尾格式，根據安裝的環境選擇

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/>

☐ Only show new options Back Next Cancel

設定使用的終端模擬器

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 `wintty` 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/>

☐ Only show new options

設定 git pull 指令執行的方式

Choose the default behavior of `git pull`
What should `git pull` do by default?



☒ **Default (fast-forward or merge)**
This is the standard behavior of `git pull`: fast-forward the current branch to 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/>

☐ Only show new options

安裝憑證協助儲存及管理工具，建議安裝，後續可省去輸入帳號、密碼的步驟

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/>

☐ Only show new options

額外設定，可以選第一項，增加效能

<http://rogerdudler.github.io/git-guide/>
<https://w3c.hexschool.com/git/cfdbd310>

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.

<https://gitforwindows.org/>

☐ Only show new options

Back

Next

Cancel

這兩個新功能暫時不選

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.

<https://gitforwindows.org/>

☐ Only show new options

Back

Install

Cancel