

Installing Git on Windows

Procedure:

Please note: You will need administrator's rights to install the Git software. If you do not have Administrator's right, please contact your computer support team.

1. Navigate to <http://git-scm.com/>

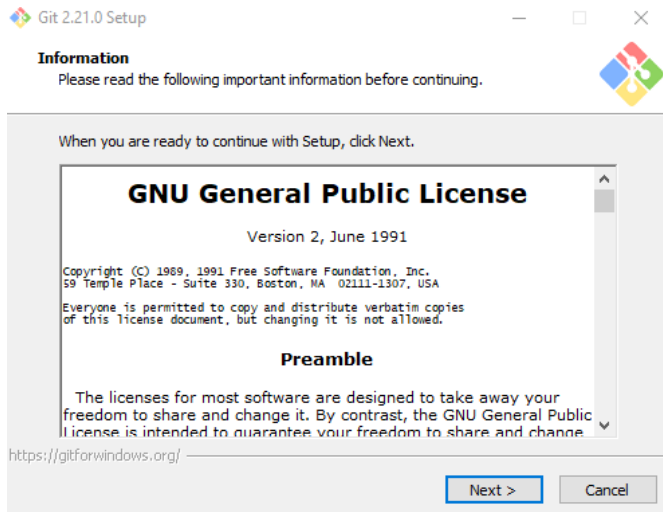
The webpage will auto-detect what operating system. Look for the computer screen on the right-hand side of the page. Click on Download 2.21.0 for Windows. Your computer should automatically launch an explorer window and ask you where to save the program on your computer.



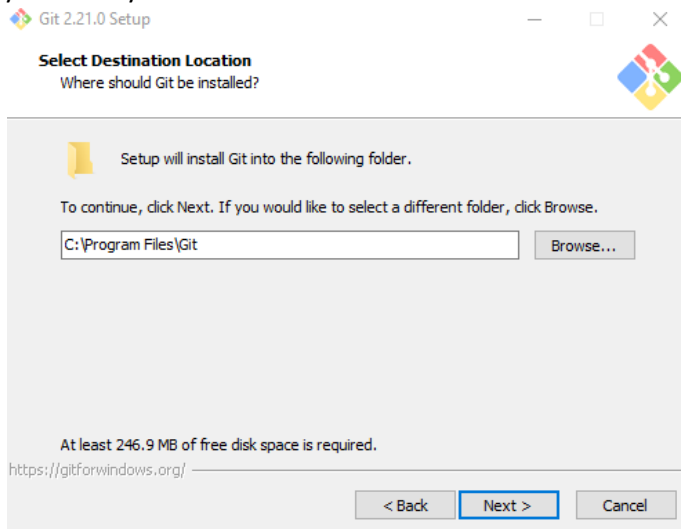
2. Navigate to where you saved the Git program .exe file. Right-click on the file and choose 'Run as Administrator'. If asked, enter in administrator credentials. This should launch the installer.

You may receive a security warning about the software being from an unknown publisher. You can safely ignore this message. Click the Run or Yes button to continue or cancel to exit out of the installation.

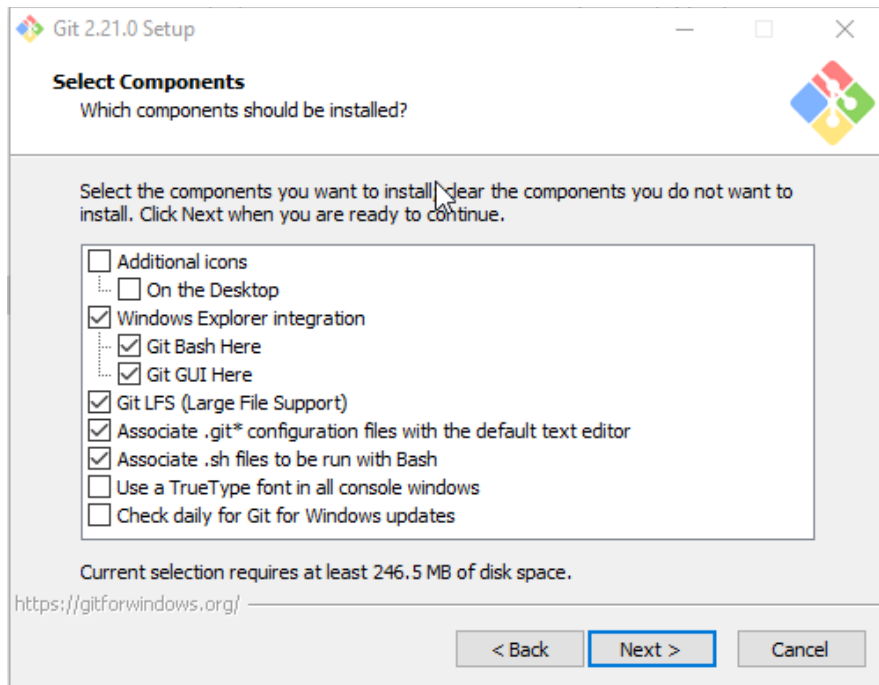
3. The Git License will display. Read through and then Click Next to accept the terms of the license.



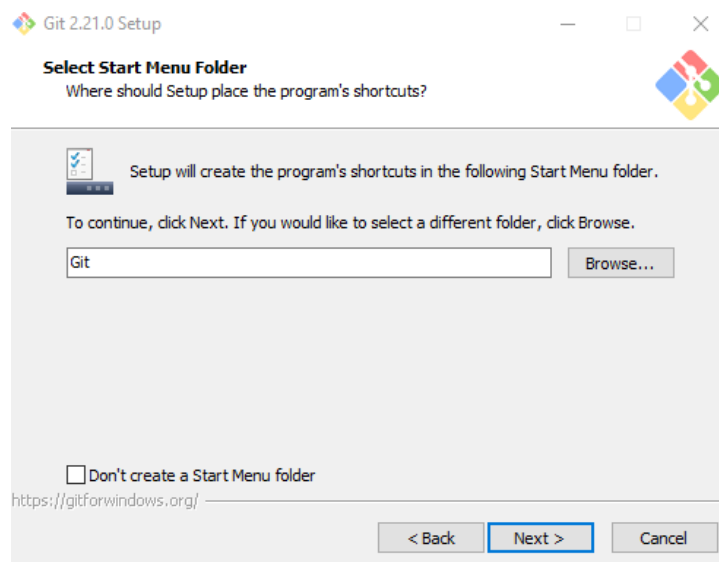
4. Choose where Git will be installed. Click Next to choose the default location provided unless you know you need to install the software somewhere else on your machine.



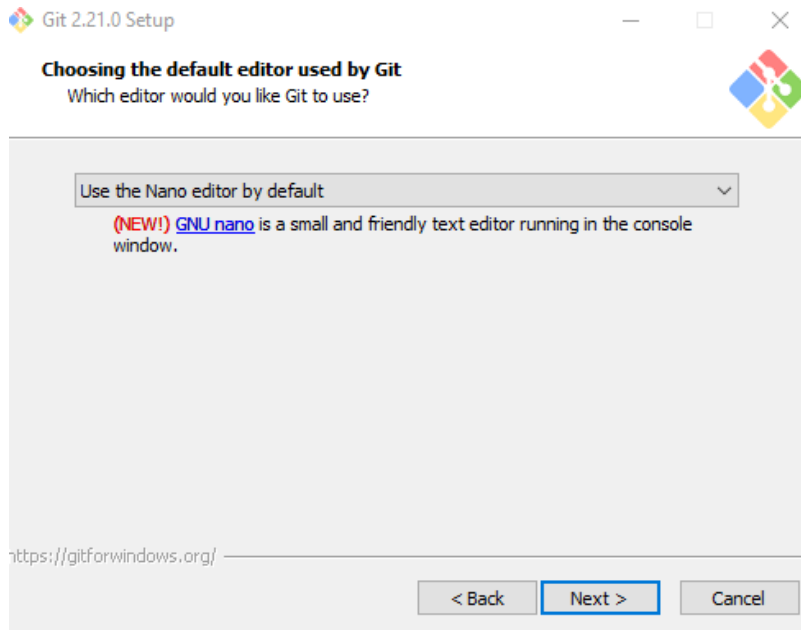
5. Select the components you want installed along with the Git software. There are already some default options checked. Make sure to check: Windows Explorer integration, including Git Bash Here and Git GUI Here. This component will allow you to use the Windows Explorer context menu to access git Bash and Git Gui. All other options are optional. See screenshot below.



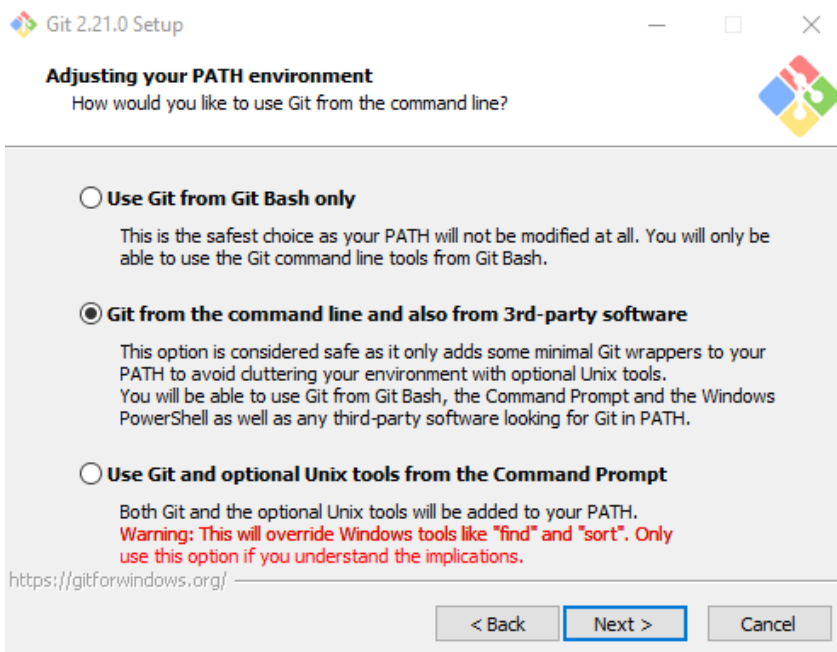
6. Select your Start menu Folder. Choose Next to use the default.



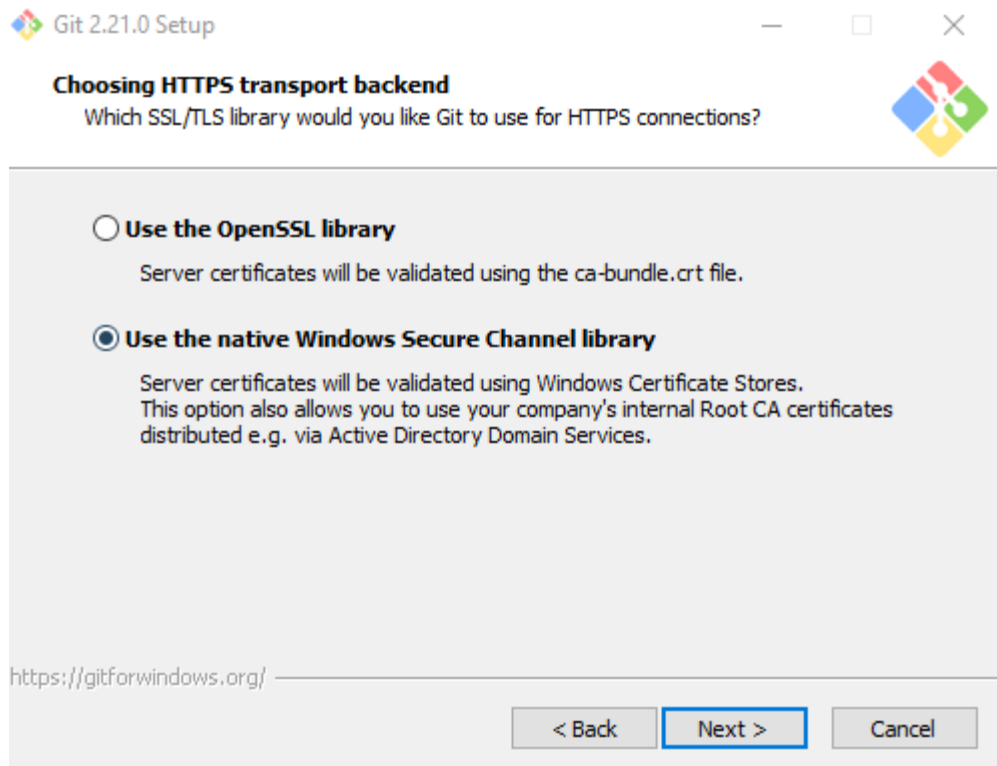
7. Choose which default editor you would like Git to use. For this class, choose 'Use the Nano editor by default' and click Next



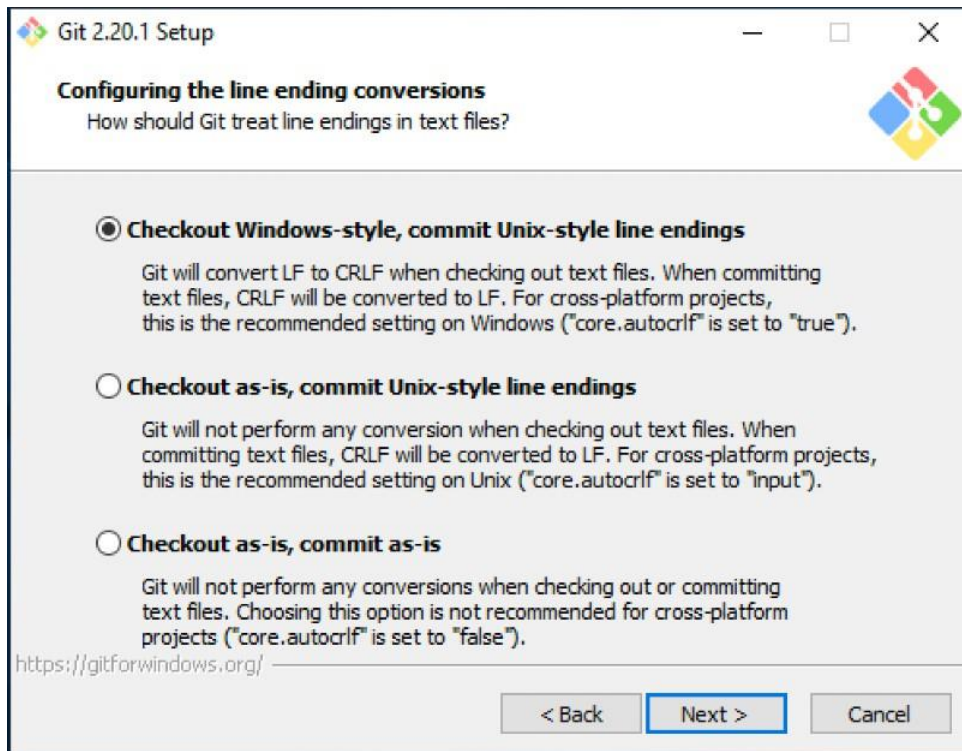
8. Choose how you would like to use Git from the command line. You can choose 'Git from the command line and also from 3rd-party software' or 'Use Git from Git Bash only' See screenshot.



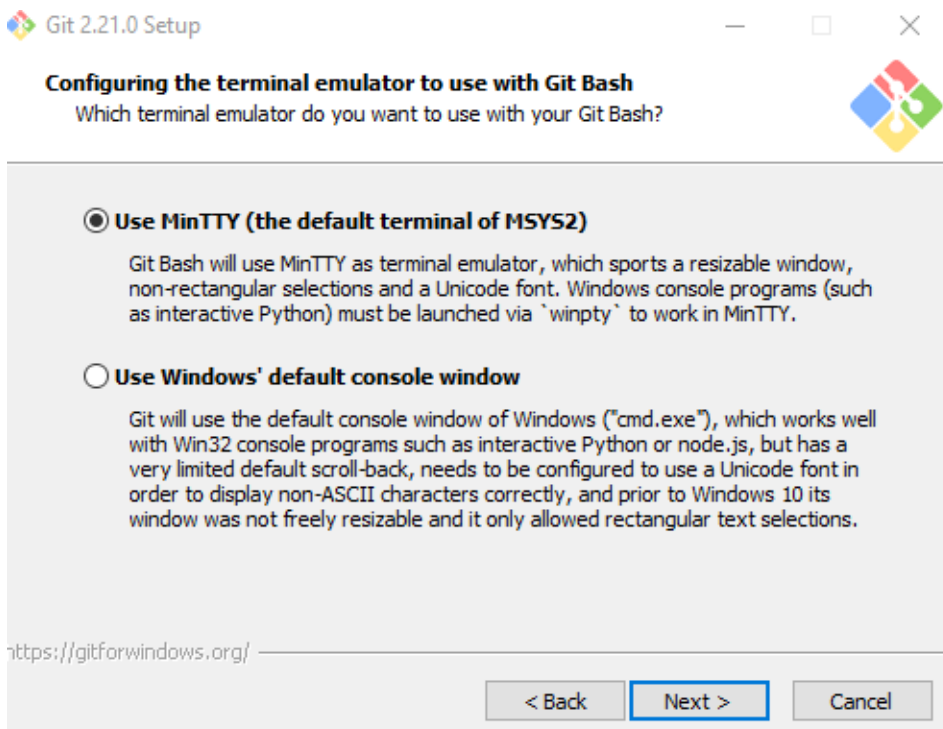
9. Choose a SSL/TLS library. Select 'Use the native Windows Secure Channel library. This allows Git to use the certificates that are native to your machine and may avoid a path problem later on. See screenshot.



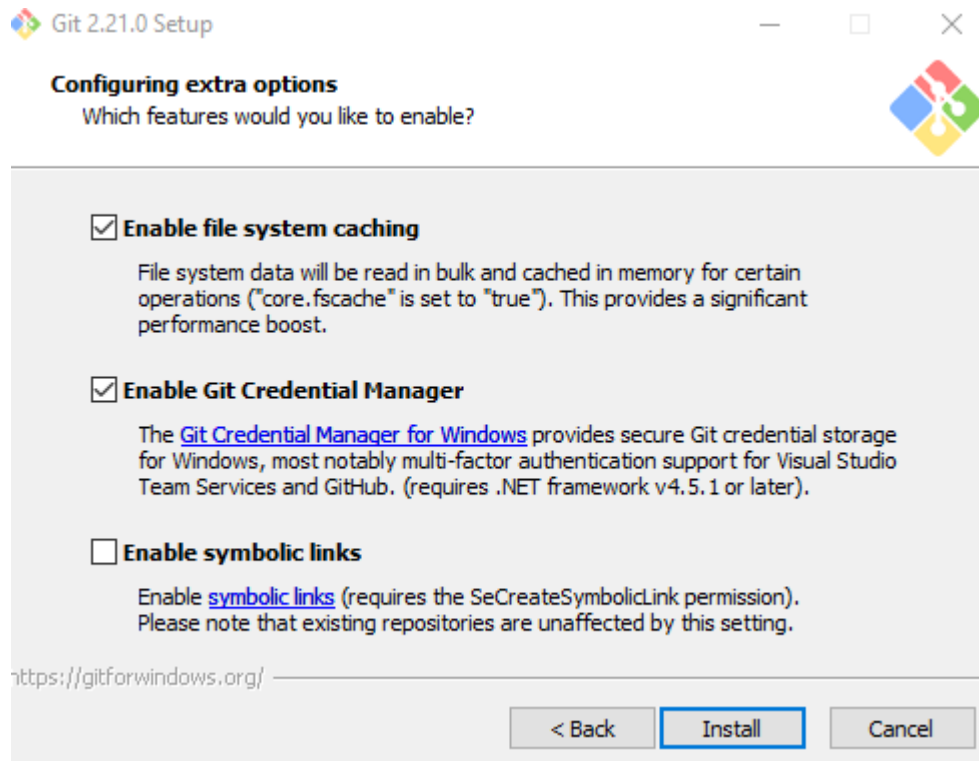
10. Choose an option for how Git should treat line endings in text files. Recommended to keep the default as this is better if you're planning on sharing your project with others who may be using a different operating system. Choose 'Checkout Windows-style, commit Unix-style line endings'. See screenshot.



11. Configure your terminal emulator to use with Git Bash. Use the default 'Use MinTTY'



12. Configure extra options. Check: Enable file system caching and Enable Git Credential Manager



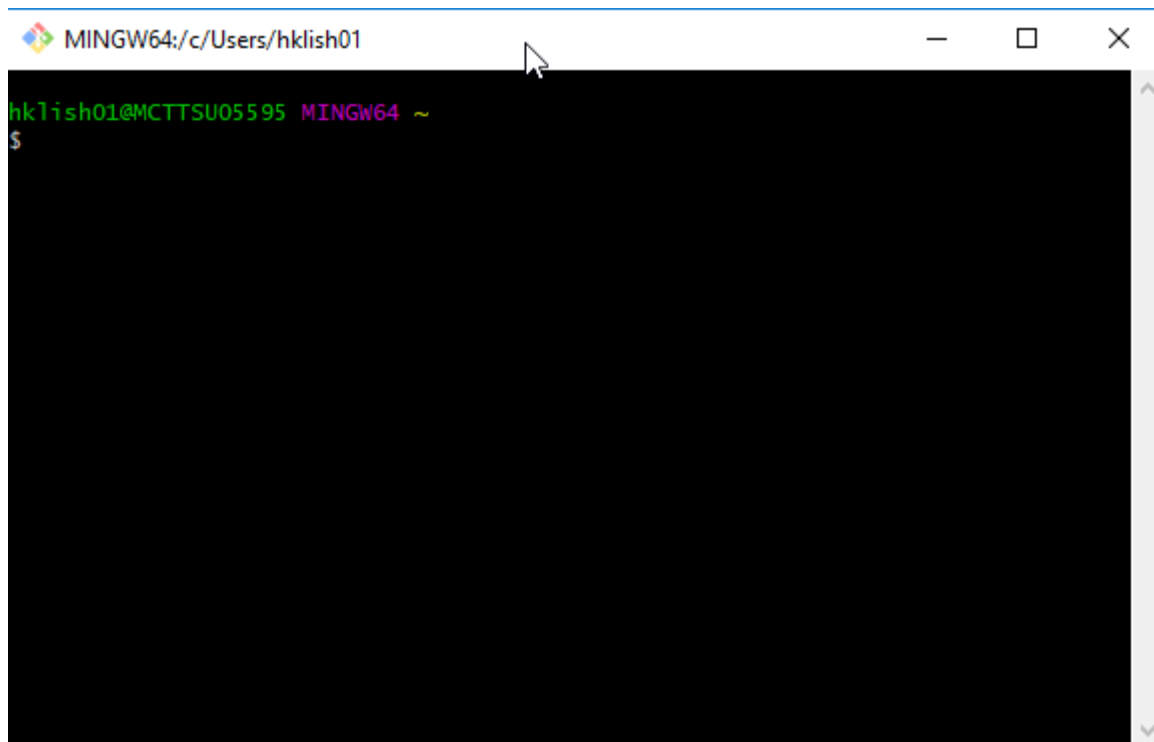
13. Choose Install to start the installation.

14. Check or uncheck the boxes to launch Git Bash or to view the release notes. Click Finish to exit the installer.

Once installed, you can access Git bash or the Git GUI via two different ways:

1. Open the Start menu, you'll find a new Git entry with the Git Bash icon
2. Best Way: Navigate to a folder on your machine where you want to use Git
 - a. Right click inside the folder. You'll notice two new options in your context menu: Git Bash and Git Gui . Choose Git Bash.

Either way, you will see the following when you open Git Bash.



A screenshot of a MINGW64 terminal window. The title bar shows the path "MINGW64:/c/Users/hklish01". The terminal content shows the prompt "hklish01@MCTTSU05595 MINGW64 ~" followed by a dollar sign "\$" on a new line.

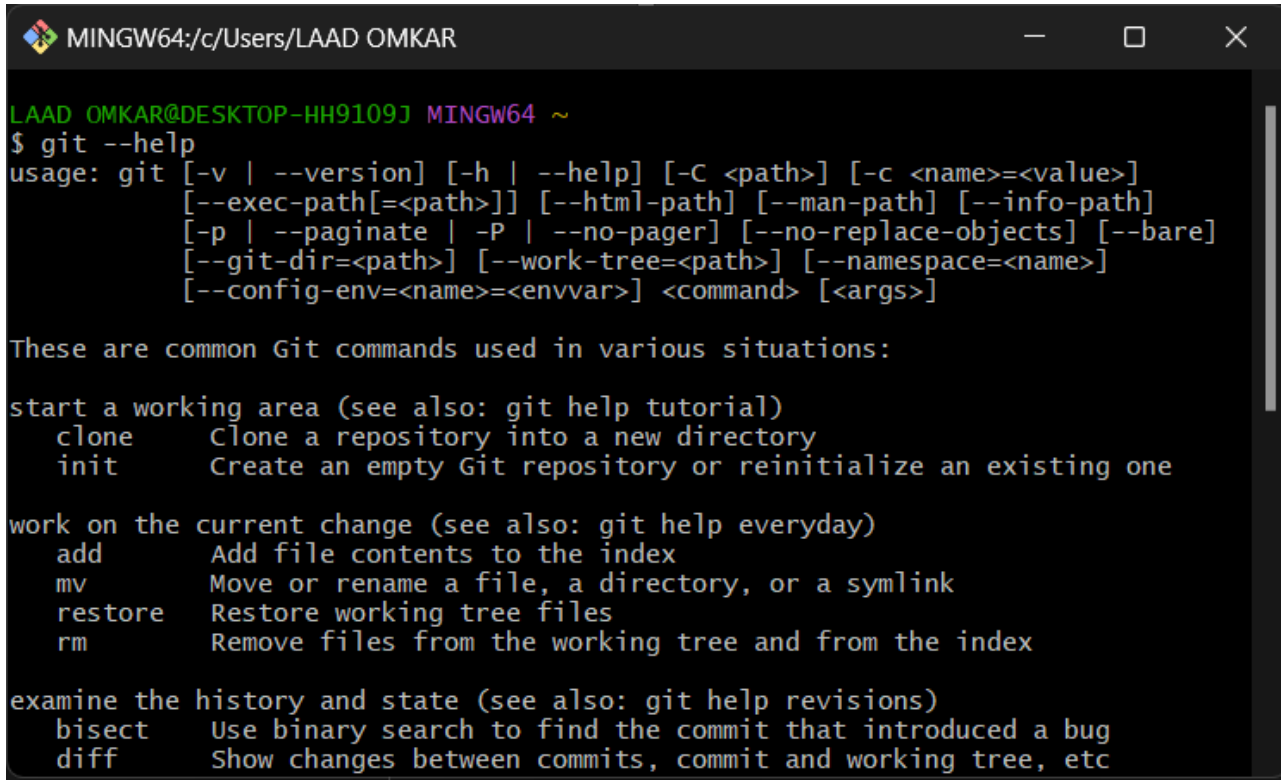
Git bash three basic commands:

1. `git --version` : This command gives you the current version of git installed in your desktop system.



A screenshot of a MINGW64 terminal window. The title bar shows the path "MINGW64:/c/Users/LAAD OMKAR". The terminal content shows the prompt "LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~", the command "\$ git --version", and the output "git version 2.43.0.windows.1". Below this, the prompt "LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~" is shown again followed by a dollar sign "\$" on a new line.

2. `git --help`: This command opens the help for understanding and knowing about the commands which can be used in git.



```
MINGW64:/c/Users/LAAD OMKAR

LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~
$ git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--config-env=<name>=<envvar>] <command> [<args>]

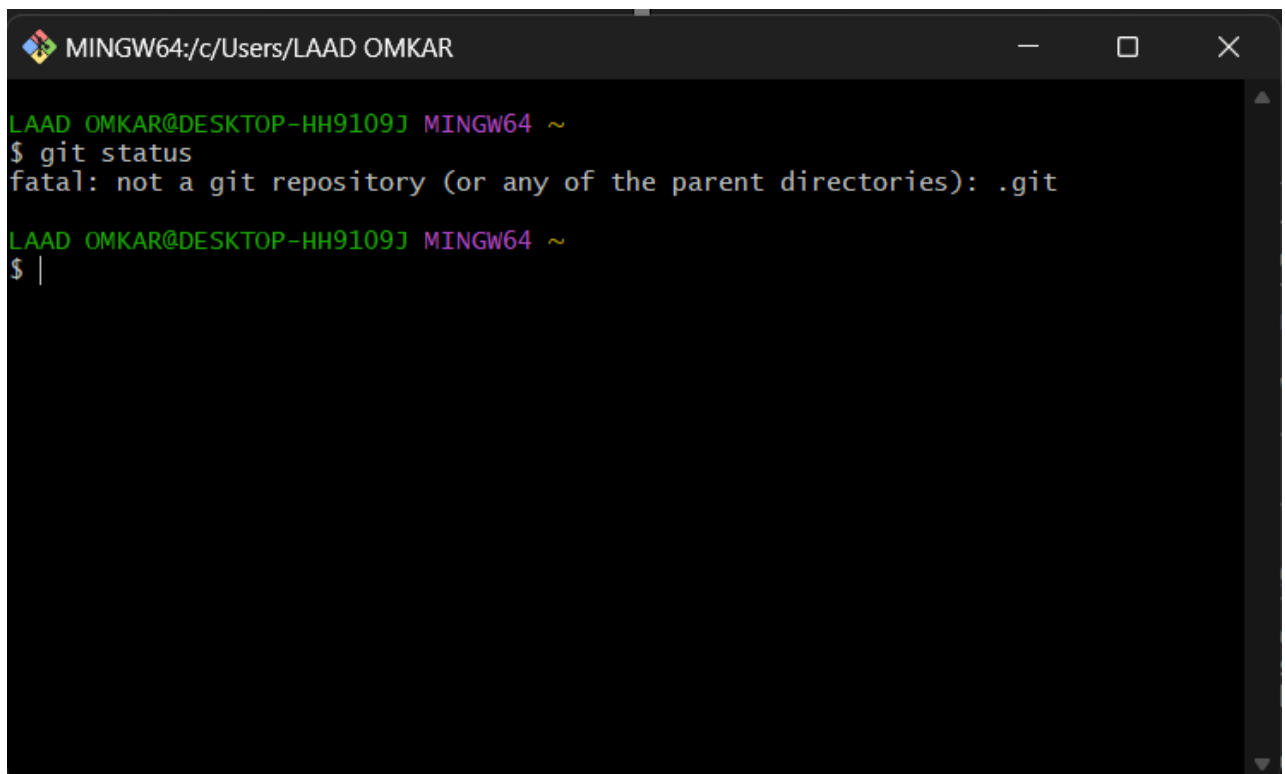
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
```

3. `git status` : This command gives you the status of you git commit.

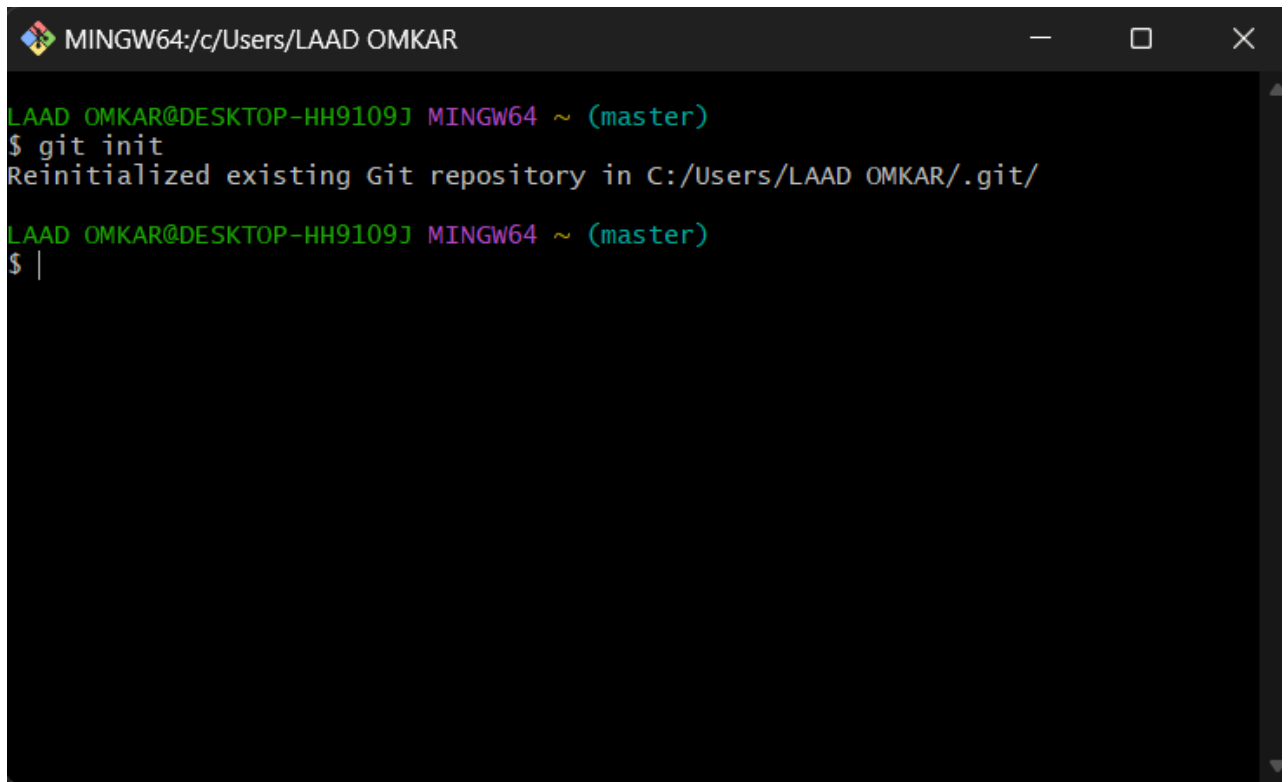


```
MINGW64:/c/Users/LAAD OMKAR

LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~
$ git status
fatal: not a git repository (or any of the parent directories): .git

LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~
$ |
```

4. `git init` : This command is used to initialize the repository.



```
MINGW64:/c/Users/LAAD OMKAR
LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/LAAD OMKAR/.git/
LAAD OMKAR@DESKTOP-HH9109J MINGW64 ~ (master)
$ |
```