Dell@DESKTOP-KAMCMGH MINGW64 /d/san

$ ls

1Videomaker/ 5Paisa/ Alarms/ amap/

1Videoshow/ A\_DownloaderForInstagram/ Android/

Dell@DESKTOP-KAMCMGH MINGW64 /d/san

$ cd d

bash: cd: d: No such file or directory

Dell@DESKTOP-KAMCMGH MINGW64 /d/san

$ cd /d

Dell@DESKTOP-KAMCMGH MINGW64 /d

$ cd San/5Paisa/

Dell@DESKTOP-KAMCMGH MINGW64 /d/San/5Paisa

$ where cd

INFO: Could not find files for the given pattern(s).

Dell@DESKTOP-KAMCMGH MINGW64 /d/San/5Paisa

$ where clear

C:\Program Files\Git\usr\bin\clear.exe

Dell@DESKTOP-KAMCMGH MINGW64 /d/San/5Paisa

$ cd ..

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ cd .

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ mkdir test

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ rmdir test

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ mkdir koluppupudichava

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ rmdir koluppupudichava

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ echo hello

hello

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ echo hello >>File.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ cat file.txt

hello

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ rmdir file.txt

rmdir: failed to remove 'file.txt': Not a directory

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ rm -r file.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/San

$ cd ..

Dell@DESKTOP-KAMCMGH MINGW64 /d

$ mkdir git

Dell@DESKTOP-KAMCMGH MINGW64 /d

$ git

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>]

[--super-prefix=<path>] [--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

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

Dell@DESKTOP-KAMCMGH MINGW64 /d

$ cd git

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ echo "hello world" >>file.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config --global user.name "AsokanT"

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config --globaluser.email "ashokt1987@gmail.com"

error: unknown option `globaluser.email'

usage: git config [<options>]

Config file location

--global use global config file

--system use system config file

--local use repository config file

--worktree use per-worktree config file

-f, --file <file> use given config file

--blob <blob-id> read config from given blob object

Action

--get get value: name [value-pattern]

--get-all get all values: key [value-pattern]

--get-regexp get values for regexp: name-regex [value-pattern]

--get-urlmatch get value specific for the URL: section[.var] URL

--replace-all replace all matching variables: name value [value-pattern]

--add add a new variable: name value

--unset remove a variable: name [value-pattern]

--unset-all remove all matches: name [value-pattern]

--rename-section rename section: old-name new-name

--remove-section remove a section: name

-l, --list list all

--fixed-value use string equality when comparing values to 'value-pattern'

-e, --edit open an editor

--get-color find the color configured: slot [default]

--get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --type <type> value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--bool-or-str value is --bool or string

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --null terminate values with NUL byte

--name-only show variable names only

--includes respect include directives on lookup

--show-origin show origin of config (file, standard input, blob, command line)

--show-scope show scope of config (worktree, local, global, system, command)

--default <value> with --get, use default value when missing entry

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ ^C

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config --global user.email "ashokt1987@gmail.com"

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config --global -1

error: unknown switch `1'

usage: git config [<options>]

Config file location

--global use global config file

--system use system config file

--local use repository config file

--worktree use per-worktree config file

-f, --file <file> use given config file

--blob <blob-id> read config from given blob object

Action

--get get value: name [value-pattern]

--get-all get all values: key [value-pattern]

--get-regexp get values for regexp: name-regex [value-pattern]

--get-urlmatch get value specific for the URL: section[.var] URL

--replace-all replace all matching variables: name value [value-pattern]

--add add a new variable: name value

--unset remove a variable: name [value-pattern]

--unset-all remove all matches: name [value-pattern]

--rename-section rename section: old-name new-name

--remove-section remove a section: name

-l, --list list all

--fixed-value use string equality when comparing values to 'value-pattern'

-e, --edit open an editor

--get-color find the color configured: slot [default]

--get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --type <type> value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--bool-or-str value is --bool or string

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --null terminate values with NUL byte

--name-only show variable names only

--includes respect include directives on lookup

--show-origin show origin of config (file, standard input, blob, command line)

--show-scope show scope of config (worktree, local, global, system, command)

--default <value> with --get, use default value when missing entry

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config --global -l

user.name=AsokanT

user.email=ashokt1987@gmail.com

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git config -l

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=AsokanT

user.email=ashokt1987@gmail.com

Dell@DESKTOP-KAMCMGH MINGW64 /d/git

$ git init

Initialized empty Git repository in D:/git/.git/

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

file.txt

nothing added to commit but untracked files present (use "git add" to track)

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git add file.txt

warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: file.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git log

fatal: your current branch 'master' does not have any commits yet

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git comit -m "initial commit"

git: 'comit' is not a git command. See 'git --help'.

The most similar command is

commit

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git commit -m "initial commit"

[master (root-commit) 943a469] initial commit

1 file changed, 1 insertion(+)

create mode 100644 file.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git status

On branch master

nothing to commit, working tree clean

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git log

commit 943a469353a03bdd0fb181d4fdeb549c98bd6a7f (HEAD -> master)

Author: AsokanT <ashokt1987@gmail.com>

Date: Wed Aug 10 21:13:00 2022 +0530

initial commit

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git status

On branch master

nothing to commit, working tree clean

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ 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>]

[--super-prefix=<path>] [--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

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git help remote

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git remote add origin https://github.com/AsokanT/test.git

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git remote

origin

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ get branch -m main

bash: get: command not found

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (master)

$ git branch -M main

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git push -u Origin main

fatal: 'Origin' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git push -u origin main

fatal: helper error (-1): User cancelled dialog.

$ git push -u origin main

remote: Support for password authentication was removed on August 13, 2021.

remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.

fatal: Authentication failed for 'https://github.com/AsokanT/test.git/'

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ $ git push -u origin main

bash: $: command not found

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git push -u origin main

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 225 bytes | 6.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/AsokanT/test.git

\* [new branch] main -> main

branch 'main' set up to track 'origin/main'.

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$ git pull origin

remote: Enumerating objects: 4, done.

remote: Counting objects: 100% (4/4), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 655 bytes | 2.00 KiB/s, done.

From https://github.com/AsokanT/test

943a469..afe68a4 main -> origin/main

Updating 943a469..afe68a4

Fast-forward

dummy | 1 +

1 file changed, 1 insertion(+)

create mode 100644 dummy

Dell@DESKTOP-KAMCMGH MINGW64 /d/git (main)

$

Initiate git for New project

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project

$ git init

Initialized empty Git repository in D:/Learning/git/new project/.git/

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

Index.html

readme.txt

nothing added to commit but untracked files present (use "git add" to track)

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project (master)

$ git add .

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project (master)

$ git commit -m "initial commit"

[master (root-commit) 8012bdf] initial commit

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 Index.html

create mode 100644 readme.txt

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project (master)

$ git status

On branch master

nothing to commit, working tree clean

Dell@DESKTOP-KAMCMGH MINGW64 /d/Learning/git/new project (master)

$