

```
H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
```

```
$ 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>]
          [--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.

```
H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
```

```
$ git init
Initialized empty Git repository in C:/Users/H-CSE-407-28/Desktop/se/.git/
```

```
H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
```

```
$ git congif --globaluser.name "navyashree"
git: 'congif' is not a git command. See 'git --help'.
```

The most similar command is

config

```
H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
```

```
$ git config --globaluser.name "navyashree"
error: unknown option 'globaluser.name'
usage: git config [<options>]
```

Config file location

--[no-]global	use global config file
--[no-]system	use system config file
--[no-]local	use repository config file
--[no-]worktree	use per-worktree config file
-f, --[no-]file <file>	use given config file
--[no-]blob <blob-id>	read config from given blob object

Action

--[no-]get	get value: name [value-pattern]
--[no-]get-all	get all values: key [value-pattern]
--[no-]get-regexp	get values for regexp: name-regex [value-pattern]

```

--[no-]get-urlmatch    get value specific for the URL: section[.var] URL
--[no-]replace-all    replace all matching variables: name value [value-
pattern]
--[no-]add              add a new variable: name value
--[no-]unset            remove a variable: name [value-pattern]
--[no-]unset-all       remove all matches: name [value-pattern]
--[no-]rename-section   rename section: old-name new-name
--[no-]remove-section   remove a section: name
-l, --[no-]list        list all
--[no-]fixed-value      use string equality when comparing values to 'value-
pattern'
-e, --[no-]edit         open an editor
--[no-]get-color        find the color configured: slot [default]
--[no-]get-colorbool    find the color setting: slot [stdout-is-tty]

```

Type

```

-t, --[no-]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, --[no-]null        terminate values with NUL byte
--[no-]name-only        show variable names only
--[no-]includes         respect include directives on lookup
--[no-]show-orig-in     show origin of config (file, standard input, blob,
command line)
--[no-]show-scope       show scope of config (worktree, local, global, system,
command)
--[no-]default <value> with --get, use default value when missing entry

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ git config --globaluser.email "navyashreepatilolla@gmail.com"
error: unknown option `globaluser.email'
usage: git config [<options>]

```

Config file location

```

--[no-]global          use global config file
--[no-]system          use system config file
--[no-]local           use repository config file
--[no-]worktree        use per-worktree config file
-f, --[no-]file <file> use given config file
--[no-]blob <blob-id>  read config from given blob object

```

Action

```

--[no-]get             get value: name [value-pattern]
--[no-]get-all        get all values: key [value-pattern]
--[no-]get-regex       get values for regexp: name-regex [value-pattern]
--[no-]get-urlmatch    get value specific for the URL: section[.var] URL
--[no-]replace-all    replace all matching variables: name value [value-
pattern]
--[no-]add              add a new variable: name value
--[no-]unset            remove a variable: name [value-pattern]
--[no-]unset-all       remove all matches: name [value-pattern]
--[no-]rename-section   rename section: old-name new-name
--[no-]remove-section   remove a section: name
-l, --[no-]list        list all
--[no-]fixed-value      use string equality when comparing values to 'value-
pattern'
-e, --[no-]edit         open an editor
--[no-]get-color        find the color configured: slot [default]
--[no-]get-colorbool    find the color setting: slot [stdout-is-tty]

```

Type

```

-t, --[no-]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, --[no-]null       terminate values with NUL byte
--[no-]name-only      show variable names only
--[no-]includes       respect include directives on lookup
--[no-]show-origins   show origin of config (file, standard input, blob,
command line)
--[no-]show-scope     show scope of config (worktree, local, global, system,
command)
--[no-]default <value>
                        with --get, use default value when missing entry

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ git clone 'https://github.com/navyashree444/sample-git.git'
Cloning into 'sample-git'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 19 (delta 2), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (19/19), done.
Resolving deltas: 100% (2/2), done.

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ ls
sample-git/

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ cd sample-git

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ git add .

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ cd ..

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ git add .
warning: adding embedded git repository: sample-git
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> sample-git
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached sample-git
hint:
hint: See "git help submodule" for more information.

```

```

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ git commit -m "
> first update
> "
[master (root-commit) 2813f9c] first update
Committer: CSELAB-407-28 <H-CSE-407-28@itam.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
create mode 160000 sample-git
```

```
H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
```

```
$ git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'origin'

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ ls
sample-git/

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se (master)
$ cd sample-git

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ git commit -m "
> second update
> "
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ git push origin main
Everything up-to-date

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ git pull origin main
From https://github.com/navyashree444/sample-git
* branch          main          -> FETCH_HEAD
Already up to date.

H-CSE-407-28@H-B-CSE-L407-28 MINGW64 ~/Desktop/se/sample-git (main)
$ ^C

H-CSE-407-28@
```