

```
MINGW64/c/Users/hasso/Desktop/git/learn_git_again
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git
$ mkdir learn_git_again
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git
$ cd learn_git_again
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again
$ touch third.txt
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again
$ git init
Initialized empty Git repository in C:/Users/hasso/Desktop/git/learn_git_again/.git/
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git add "third.txt"
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git commit -m "commit third.txt"
[master (root-commit) 1ba09de] commit third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third.txt
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git log
commit 1ba09de2f0b149d7Fe138908bd38f32fcb5c2330 (HEAD -> master)
Author: houssem-mbarki <hassou2395@gmail.com>
Date:   Wed Sep 8 10:15:01 2021 +0100

    commit third.txt
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ touch fourth.txt
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git add "fourth.txt"
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git commit -m "adding Fourth.txt"
bash: gitcommit: command not found
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git rm "third.txt"
rm 'third.txt'
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.
```

```
MINGW64/c/Users/hasso/Desktop/git/learn_git_again
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
  add
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git add .
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git commit -m "removing third.txt"
[master 80b48d7] removing third.txt
1 file changed, 0 insertions(+), 0 deletions(-)
rename third.txt => fourth.txt (100%)
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git log
commit 80b48d703dcbb3055884d0dc3d1d55f7bdc1c3c (HEAD -> master)
Author: houssem-mbarki <hassou2395@gmail.com>
Date:   Wed Sep 8 10:25:48 2021 +0100

    removing third.txt

commit 1ba09de2f0b149d7Fe138908bd38f32fcb5c2330
Author: houssem-mbarki <hassou2395@gmail.com>
Date:   Wed Sep 8 10:15:01 2021 +0100

    commit third.txt
hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git commit --global core.pager cat
error: unknown option 'global'
usage: git commit [<options>] [--] <pathspec>...

    -q, --quiet           suppress summary after successful commit
    -v, --verbose         show diff in commit message template

Commit message options
    -F, --file <file>    read message from file
    --author <author>    override author for commit
    --date <date>        override date for commit
    -m, --message <message>
                        commit message
    -c, --reedit-message <commit>
                        reuse and edit message from specified commit
    -C, --reuse-message <commit>
                        reuse message from specified commit
    --fixup [(amend|reword):]commit
                        use autosquash formatted message to fixup or amend/reword
```

```
MINGW64/c/Users/hasso/Desktop/git/learn_git_again
--reuse-message <commit> reuse and edit message from specified commit
-C, --reuse-message <commit> reuse message from specified commit
--fixup [(amend|reword):]<commit> use autosquash formatted message to fixup or amend/reword specified commit
--squash <commit> use autosquash formatted message to squash specified commit
--reset-author the commit is authored by me now (used with -C/-c/--amend)
--trailer <trailer> add custom trailer(s)
-s, --signoff add a Signed-off-by trailer
-t, --template <file> use specified template file
-e, --edit force edit of commit
--cleanup <mode> how to strip spaces and #comments from message
--status include status in commit message template
-S, --pgp-sign[=<key-id>] GPG sign commit

Commit contents options
-a, --all commit all changed files
-i, --include add specified files to index for commit
--interactive interactively add files
-p, --patch interactively add changes
-o, --only commit only specified files
-n, --no-verify bypass pre-commit and commit-msg hooks
--dry-run show what would be committed
--short show status concisely
--branch show branch information
--ahead-behind compute full ahead/behind values
--porcelain machine-readable output
--long show status in long format (default)
-z, --null terminate entries with NUL
--amend amend previous commit
--no-post-rewrite bypass post-rewrite hook
-u, --untracked-files[=<mode>] show untracked files, optional modes: all, normal, no.
(Default: all)
--pathspec-from-file <file> read pathspec from file
--pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character

hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git config --global core.pager cat

hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git config --get-all
```

```
MINGW64/c/Users/hasso/Desktop/git/learn_git_again
-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-regexp [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-ty]

Type
-t, --type <v> 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

hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ git config --global --list
user.email=hasso23955@gmail.com
user.name=houssen-mbarki
core.pager=cat

hasso@LAPTOP-MB41L7PQ MINGW64 ~/Desktop/git/learn_git_again (master)
$ |
```