Directory check –pwd

Version – git --version

Configuration- git config --g user.name “Ishan”

git config --g user.email “ishan19798@gmail.com”

creating rep- git init

joining rep- git clone <url>

status – git status

commit -$ git commit -m "wi1:file created hello world"

$ git commit -am "wi1:file updated"

For untracked file use – git add .

Update history – git log

Update filter - git log -2

$ git log --since="09/22/2020"

$ git log --author ="ishan" --oneline

$ git log --until="09/23/2020"

$ git log --before="09/23/2020"

$ git log --grep="wi1"

$ git log --grep="file2"

$ git log --oneline

Help- git log –help

To move head - $ git checkout 1a6473f

To move to original place - $ git checkout master

Add tag - $ git tag v1.3 313ffe7

To find changes made - $ git diff

Undo ways - $ git checkout -- deleting.txt

For uncommitted-checkout /restore

For commited changes-revert-safe undo commited changes

-resest-un safe undo commited changes

Day 1

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ pwd

/c/Users/ShreeVastav/Documents/Git

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git\

>

usage: git [--version] [--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>]

<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

sparse-checkout Initialize and modify the sparse-checkout

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.

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git --version

git version 2.28.0.windows.1

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --g user.name "Ishan"

error: ambiguous option: g (could be --get-color or --get-colorbool)

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-regex]

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

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

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

--replace-all replace all matching variables: name value [value\_regex]

--add add a new variable: name value

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

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

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

--remove-section remove a section: name

-l, --list list all

-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 <> value is given this type

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

--int value is decimal number

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

--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

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

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

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --global user.email "Ishan"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ --list

bash: --list: command not found

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config -- list

error: key does not contain a section: list

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --list

pack.packsizelimit=2g

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/mingw32/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

user.name=Ishan

user.email=Ishan

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --global user.email "ishan19798@gmail.com"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --list

pack.packsizelimit=2g

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/mingw32/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

user.name=Ishan

user.email=ishan19798@gmail.com

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --global core.editor notepad

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git config --list

pack.packsizelimit=2g

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/mingw32/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

user.name=Ishan

user.email=ishan19798@gmail.com

core.editor=notepad

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git clone <url>

bash: syntax error near unexpected token `newline'

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git

$ git init

Initialized empty Git repository in C:/Users/ShreeVastav/Documents/Git/.git/

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

No commits yet

Untracked files:

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

New Text Document.txt

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

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git add

Nothing specified, nothing added.

hint: Maybe you wanted to say 'git add .'?

hint: Turn this message off by running

hint: "git config advice.addEmptyPathspec false"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git add .

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git commit -m "wi1:file created hello world"

[master (root-commit) 1a6473f] wi1:file created hello world

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 New Text Document.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

nothing to commit, working tree clean

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: New Text Document.txt

no changes added to commit (use "git add" and/or "git commit -a")

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git commit -am "wi1:file updated"

[master 313ffe7] wi1:file updated

1 file changed, 1 insertion(+)

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Untracked files:

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

file 2.txt

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

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git add .

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git commit^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git c

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

The most similar commands are

clone

commit

gc

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: file 2.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: file 2.txt

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: file 2.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git commit

hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/ShreeVastav/Documents/Git/.git/COMMIT\_EDITMSG to DOS format...

dos2unix: converting file C:/Users/ShreeVastav/Documents/Git/.git/COMMIT\_EDITMSG to Unix format...

[master 5747fbc] haha

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 file 2.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log

commit 5747fbc86d0d95ebec8ff95ba233cc989937bfbd (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:47:49 2020 +0530

haha

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log -2

commit 5747fbc86d0d95ebec8ff95ba233cc989937bfbd (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:47:49 2020 +0530

haha

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --author ="Ishan" --oneline

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --author ="ishan" --oneline

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --author ="Ishan"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --since="09/22/2020"

commit 5747fbc86d0d95ebec8ff95ba233cc989937bfbd (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:47:49 2020 +0530

haha

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --untill="09/23/2020"

fatal: unrecognized argument: --untill=09/23/2020

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --until="09/23/2020"

commit 5747fbc86d0d95ebec8ff95ba233cc989937bfbd (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:47:49 2020 +0530

haha

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --before="09/23/2020"

commit 5747fbc86d0d95ebec8ff95ba233cc989937bfbd (HEAD -> master)

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:47:49 2020 +0530

haha

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --grep="wi1"

commit 313ffe7cf394c3bf9c9668e826e0d9189dc5c1cc

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:42:02 2020 +0530

wi1:file updated

commit 1a6473f05fceb2834cbdf77fb142b9c22125065b

Author: Ishan <ishan19798@gmail.com>

Date: Wed Sep 23 12:39:45 2020 +0530

wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --grep="File2"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --grep="File1"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --grep="file2"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --help

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

5747fbc (HEAD -> master) haha

313ffe7 wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: file 2.txt

Untracked files:

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

deleting.txt

no changes added to commit (use "git add" and/or "git commit -a")

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git add .

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git commit -m "file to be deleted"

[master 1c07b06] file to be deleted

2 files changed, 1 insertion(+)

create mode 100644 deleting.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

nothing to commit, working tree clean

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

deleted: deleting.txt

no changes added to commit (use "git add" and/or "git commit -a")

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git rm

fatal: No pathspec was given. Which files should I remove?

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git rm deleting

fatal: pathspec 'deleting' did not match any files

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git rm file2

fatal: pathspec 'file2' did not match any files

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git rm file2.txt

fatal: pathspec 'file2.txt' did not match any files

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git checkout 1a6473f

Note: switching to '1a6473f'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git ((1a6473f...))

$ git log --oneline

1a6473f (HEAD) wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git ((1a6473f...))

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git ((1a6473f...))

$ git checkout master

Previous HEAD position was 1a6473f wi1:file created hello world

Switched to branch 'master'

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git tag

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git tag v1.3 313ffe7

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 (tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git diff

**diff --git a/deleting.txt b/deleting.txt**

**index e69de29..c0b8247 100644**

**--- a/deleting.txt**

**+++ b/deleting.txt**

@@ -0,0 +1,8 @@

+nvjvnkd

+gf

+fh

+gfh

+gf

+hgf

+j

+gfj

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: deleting.txt

no changes added to commit (use "git add" and/or "git commit -a")

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git checkout --deleting.txt

error: unknown option `deleting.txt'

usage: git checkout [<options>] <branch>

or: git checkout [<options>] [<branch>] -- <file>...

-b <branch> create and checkout a new branch

-B <branch> create/reset and checkout a branch

-l create reflog for new branch

--guess second guess 'git checkout <no-such-branch>' (default)

--overlay use overlay mode (default)

-q, --quiet suppress progress reporting

--recurse-submodules[=<checkout>]

control recursive updating of submodules

--progress force progress reporting

-m, --merge perform a 3-way merge with the new branch

--conflict <style> conflict style (merge or diff3)

-d, --detach detach HEAD at named commit

-t, --track set upstream info for new branch

-f, --force force checkout (throw away local modifications)

--orphan <new-branch>

new unparented branch

--overwrite-ignore update ignored files (default)

--ignore-other-worktrees

do not check if another worktree is holding the given ref

-2, --ours checkout our version for unmerged files

-3, --theirs checkout their version for unmerged files

-p, --patch select hunks interactively

--ignore-skip-worktree-bits

do not limit pathspecs to sparse entries only

--pathspec-from-file <file>

read pathspec from file

--pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git restore <file>..." to discard changes in working directory)

modified: deleting.txt

no changes added to commit (use "git add" and/or "git commit -a")

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git checkout -- deleting.txt

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git status

On branch master

nothing to commit, working tree clean

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git revert 313ffe7

hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/ShreeVastav/Documents/Git/.git/COMMIT\_EDITMSG to DOS format...

dos2unix: converting file C:/Users/ShreeVastav/Documents/Git/.git/COMMIT\_EDITMSG to Unix format...

[master b7199c0] Revert "wi1:file updated"

1 file changed, 1 deletion(-)

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

b7199c0 (HEAD -> master) Revert "wi1:file updated"

1c07b06 file to be deleted

5747fbc haha

313ffe7 (tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git reset --hardb7199c0

error: unknown option `hardb7199c0'

usage: git reset [--mixed | --soft | --hard | --merge | --keep] [-q] [<commit>]

or: git reset [-q] [<tree-ish>] [--] <pathspec>...

or: git reset [-q] [--pathspec-from-file [--pathspec-file-nul]] [<tree-ish>]

or: git reset --patch [<tree-ish>] [--] [<pathspec>...]

or: DEPRECATED: git reset [-q] [--stdin [-z]] [<tree-ish>]

-q, --quiet be quiet, only report errors

--mixed reset HEAD and index

--soft reset only HEAD

--hard reset HEAD, index and working tree

--merge reset HEAD, index and working tree

--keep reset HEAD but keep local changes

--recurse-submodules[=<reset>]

control recursive updating of submodules

-p, --patch select hunks interactively

-N, --intent-to-add record only the fact that removed paths will be added later

--pathspec-from-file <file>

read pathspec from file

--pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character

-z DEPRECATED (use --pathspec-file-nul instead): paths are separated with NUL character

--stdin DEPRECATED (use --pathspec-from-file=- instead): read paths from <stdin>

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git reset --hard b7199c0

HEAD is now at b7199c0 Revert "wi1:file updated"

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

b7199c0 (HEAD -> master) Revert "wi1:file updated"

1c07b06 file to be deleted

5747fbc haha

313ffe7 (tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git reset --hard 1c07b06\

>

HEAD is now at 1c07b06 file to be deleted

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git reset --hard 1c07b06

HEAD is now at 1c07b06 file to be deleted

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 (tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git resert --hard HEAD-2

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

The most similar command is

revert

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

1c07b06 (HEAD -> master) file to be deleted

5747fbc haha

313ffe7 (tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git resert --hard HEAD~2

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

The most similar command is

revert

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git reset --hard HEAD~2

HEAD is now at 313ffe7 wi1:file updated

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ git log --oneline

313ffe7 (HEAD -> master, tag: v1.3) wi1:file updated

1a6473f wi1:file created hello world

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$ ^C

ShreeVastav@ShreeVastav-PC MINGW32 ~/Documents/Git (master)

$