

If we manually rename files in file explorer (rename second to third). Then git doesn't know about it, it thinks you created a new file until you put files in staging area

```
MINGW32 /e/videos/study/cs/important/Git and github/Notes/main
added gitignore

commit d98a84f58af5fe4ebd6df21a9086fd93f86512d1
Author: Suraj <yadavsurajkumar665@gmail.com>
Date: Mon Apr 18 18:10:06 2022 +0530

    first commit

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (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:   second.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    third.txt

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

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git add .

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
on branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   second.txt -> third.txt

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ |
```

now commit the changes

```
MINGW32 /e/videos/study/cs/important/Git and github/Notes/main
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:   second.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    third.txt

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

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git add .

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
on branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   second.txt -> third.txt

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git commit -m "rename second to third"
[master cf292bd] rename second to third
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename second.txt => third.txt (100%)

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
on branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$
```

How to rename and remove files using git?

1) Deleting a file

Type “git rm filename.extension”

```
MINGW32/e/videos/Study/cs/important/Git and github/Notes/main
$ git add .
dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   second.txt -> third.txt

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "rename second to third"
[master cf292bd] rename second to third
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename second.txt => third.txt (100%)

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git rm third.txt
rm 'third.txt'

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   third.txt

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$
```

It will automatically stage it. Now commit for this deleted file “git commit –m “message””

```
MINGW32/e/videos/Study/cs/important/Git and github/Notes/main
On branch master
nothing to commit, working tree clean

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git rm third.txt
rm 'third.txt'

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   third.txt

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "removed bad file third.txt"
[master dcf6bdd] removed bad file third.txt
 1 file changed, 2 deletions(-)
 delete mode 100644 third.txt

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git sttaus
git: 'sttaus' is not a git command. See 'git --help'.

The most similar command is
  status

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$
```

2) Rename

“git mv originalFileName.extension newFileName.extension”

```
Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main
(use "git restore --staged <file>..." to unstage)
  deleted:    third.txt

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "removed bad file third.txt"
[master dcf6bdd] removed bad file third.txt
 1 file changed, 2 deletions(-)
 delete mode 100644 third.txt

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git sttaus
git: 'sttaus' is not a git command. see 'git --help'.

The most similar command is
      status

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git mv first.txt renamedFirst.txt

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    first.txt -> renamedFirst.txt

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ |
```

It also put file in staging area. Now commit this change “git commit –m “message””

```
Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main
$ git sttaus
git: 'sttaus' is not a git command. see 'git --help'.

The most similar command is
      status

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git mv first.txt renamedFirst.txt

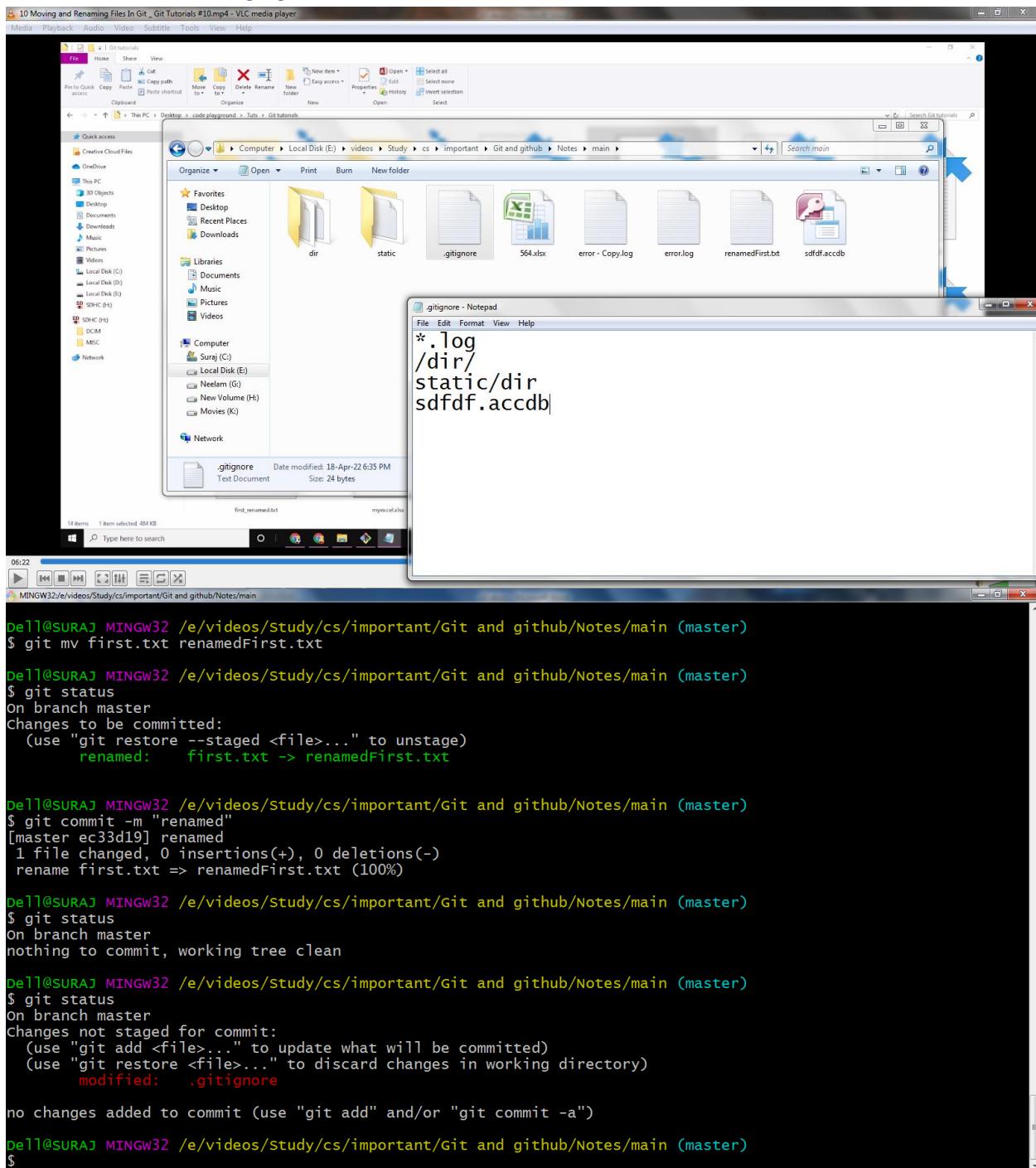
Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    first.txt -> renamedFirst.txt

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "renamed"
[master ec33d19] renamed
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename first.txt => renamedFirst.txt (100%)

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ |
```

Now add 1 file name to .gitignore file



now stage it and then commit it

```
MINGW32/e/videos/Study/cs/important/Git and github/Notes/main
$ git commit -m "renamed"
[master ec33d19] renamed
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename first.txt => renamedFirst.txt (100%)

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (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:   .gitignore

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

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git add .

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git commit -m "changed gitignore"
[master 0e2194a] changed gitignore
 1 file changed, 2 insertions(+), 1 deletion(-)

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$
```

Now here if we change the new file added in gitignore it will not work

```
MINGW32/e/videos/Study/cs/important/Git and github/Notes/main
Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (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:   .gitignore

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

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git add .

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git commit -m "changed gitignore"
[master 0e2194a] changed gitignore
 1 file changed, 2 insertions(+), 1 deletion(-)

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (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:   sdfdf.accdb

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

Dell@SURAJ MINGW32 /e/videos/study/cs/important/Git and github/Notes/main (master)
$
```

This happens because initially it was tracked by git and git will keep tracking it until we manually untracked it → use “git rm --cached filename.extension”

```
Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "changed gitignore"
[master 0e2194a] changed gitignore
 1 file changed, 2 insertions(+), 1 deletion(-)

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (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:   sdfdf.accdb

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

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git rm --cached sdfdf.accdb
rm 'sdfdf.accdb'

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   sdfdf.accdb

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ |
```

it shows deleted by it is not deleted it only untracked

Commit new change → “git commit –m “message””

```
Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (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:   sdfdf.accdb

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

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git rm --cached sdfdf.accdb
rm 'sdfdf.accdb'

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   sdfdf.accdb

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git commit -m "removed sdfdf.accdb"
[master a5a884e] removed sdfdf.accdb
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 sdfdf.accdb

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$ git status
On branch master
nothing to commit, working tree clean

Dell@SURAJ MINGW32 /e/videos/Study/cs/important/Git and github/Notes/main (master)
$
```

Now if we change sdfdf.accdb will not shown in git and since it is in .gitignore so it will also not shown in untracked section