

“GIT AND GITHUB”

SUBMITTED BY : MUNAZA ASHRAF

Version Control

Version Control is a kind of database. A version control system records the changes you make to your project's files and when you later take a look at an older snapshot (version) , your VCS shows you exactly how it differed from the previous one.. This is what version control is about. Version control is independent of the kind of project / technology / framework you're working with.

Don't confuse VCS with a backup or a deployment system. You don't have to change or replace any other part of your tool chain when you start using version control.

The most basic building block of version control is a “repository”. In Git, the repository is just a simple hidden folder named “.git” in the root directory of your project.

Why Use a Version Control System?

Collaboration

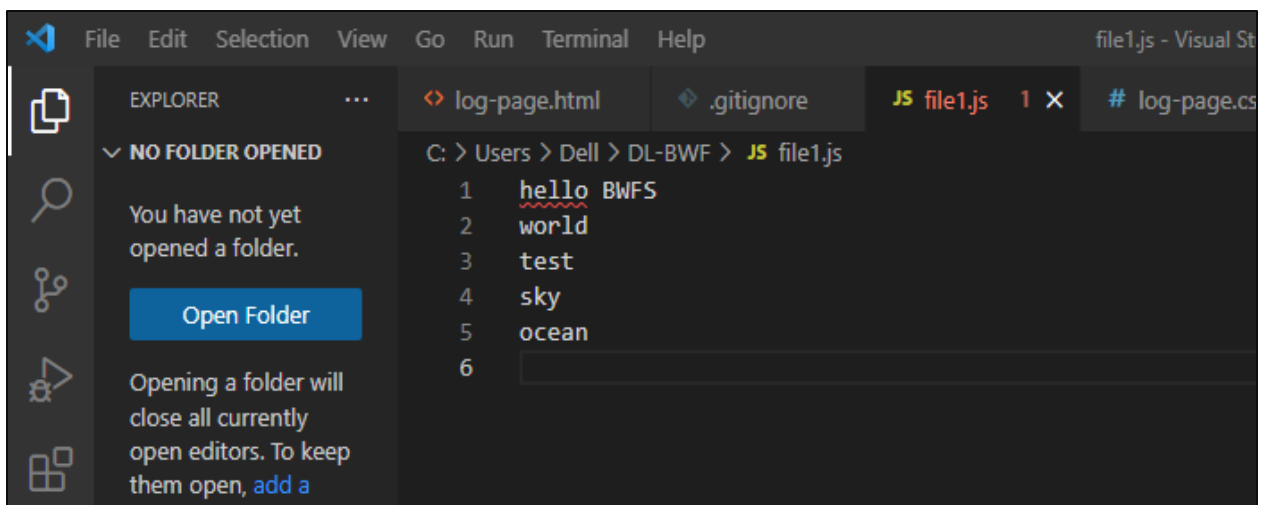
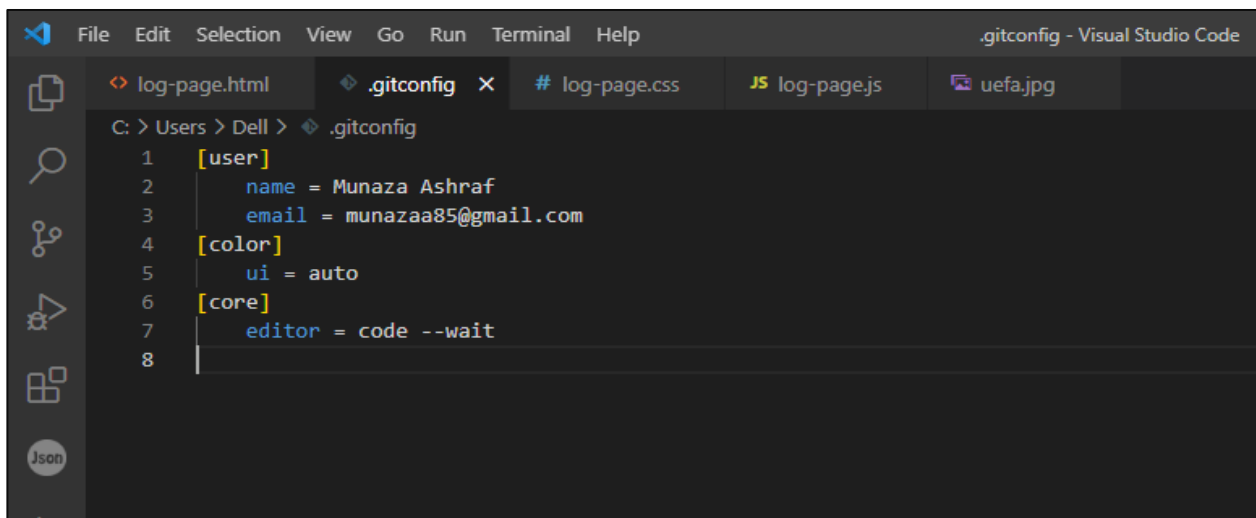
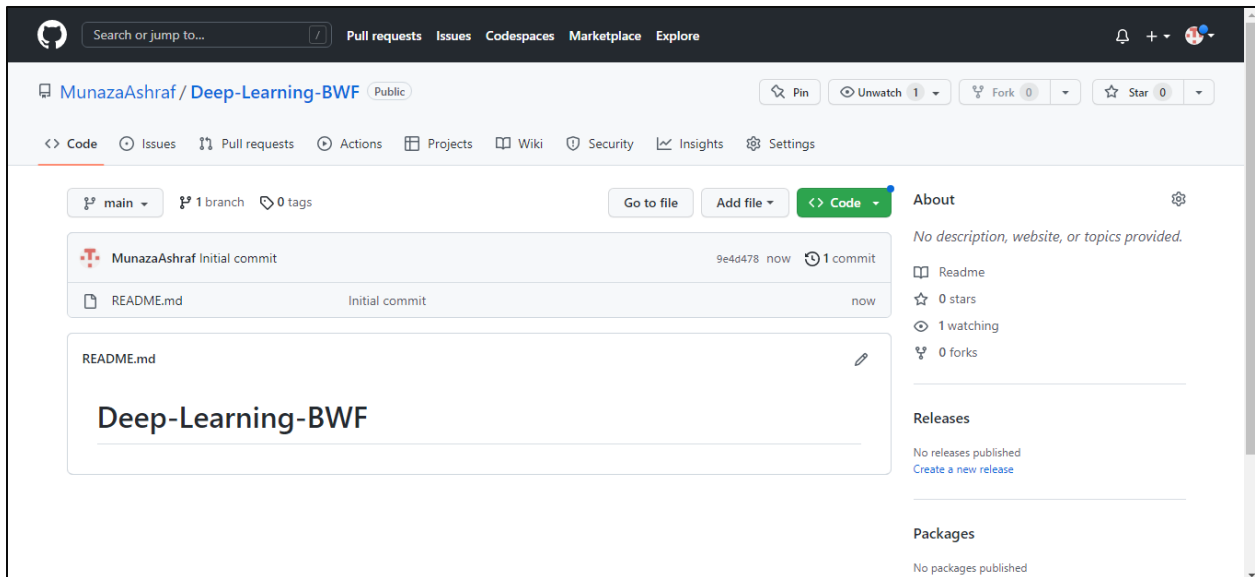
Storing Versions

Restoring Previous Versions

Understanding What Happened (Every time you save a new version of your project, your VCS requires you to provide a short description of what was changed. This helps you understand how your project evolved between version.)

Backup (every team member has a full-blown repository of the project on his disk - including the project's complete history. Should your beloved central server break down (and your backup drives fail), all you need for recovery is one of your teammates' local Git repository.)

Working



COMMANDS

```
De1l@DESKTOP-L7MC02J MINGW64 ~
$ git config --global user.name "Munaza Ashraf"

De1l@DESKTOP-L7MC02J MINGW64 ~
$ git config --global user.email munazaa85@gmail.com

De1l@DESKTOP-L7MC02J MINGW64 ~
$ code

De1l@DESKTOP-L7MC02J MINGW64 ~
$

De1l@DESKTOP-L7MC02J MINGW64 ~
$ git config --global core.editor "code --wait"

De1l@DESKTOP-L7MC02J MINGW64 ~
$ git config --global -e

De1l@DESKTOP-L7MC02J MINGW64 ~
$ git config --global core.autocrlf true

De1l@DESKTOP-L7MC02J MINGW64 ~
$ mkdir DL-BWF

De1l@DESKTOP-L7MC02J MINGW64 ~
$ cd DL-BWF

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF
$ git init
Initialized empty Git repository in C:/Users/De1l/DL-BWF/.git/

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ ls

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ ls -al
total 32
drwxr-xr-x 1 De1l 197121 0 Mar 16 19:29 ./
drwxr-xr-x 1 De1l 197121 0 Mar 16 19:28 ../
drwxr-xr-x 1 De1l 197121 0 Mar 16 19:29 .git/

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ open .git
bash: open: command not found

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ ls -a
./ ../ .git/

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ cd .git

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF/.git (GIT_DIR!)
$ cd-
bash: cd-: command not found

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF/.git (GIT_DIR!)
$ cd -
/c/Users/De1l/DL-BWF

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ echo hello > file1.txt

De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ echo hello > file2.txt
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git status  
On branch master
```

```
No commits yet
```

```
Untracked files:
```

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

```
file1.txt  
file2.txt
```

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

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git add file1.txt file2.txt
```

```
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time  
Git touches it
```

```
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time  
Git touches it
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ get status
```

```
bash: get: command not found
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

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

```
new file:   file1.txt  
new file:   file2.txt
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ echo world >> file1.txt
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

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

```
new file:   file1.txt  
new file:   file2.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:   file1.txt
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git add file1.txt
```

```
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time  
Git touches it
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

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

```
new file:   file1.txt  
new file:   file2.txt
```

```

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit
[master (root-commit) e1825e3] Initial commit.
 2 files changed, 3 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo test >> file1.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -a -m "Fix the bug that prevented users for signing up."
fatal: paths 'im ...' with -a does not make sense

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -a -m "Fix the bug."
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time
Git touches it
[master b994e3f] Fix the bug.
 1 file changed, 1 insertion(+)

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ rm file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (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:    file2.txt

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

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-files
file1.txt
file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-files
file1.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Remove unused code."
[master 9f577ec] Remove unused code.
 1 file changed, 1 deletion(-)
 delete mode 100644 file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm file2.txt
fatal: pathspec 'file2.txt' did not match any files

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo yeahh > file2.txt

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm file2.txt
fatal: pathspec 'file2.txt' did not match any files

```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file2.txt
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time
Git touches it
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm file2.txt
error: the following file has changes staged in the index:
    file2.txt
(use --cached to keep the file, or -f to force removal)
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git -f
unknown option: -f
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>]
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ -f
bash: -f: command not found
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm file2.txt
error: the following file has changes staged in the index:
    file2.txt
(use --cached to keep the file, or -f to force removal)
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-files
file1.txt
file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ rm file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file2.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-files
file1.txt
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Remved file"
On branch master
nothing to commit, working tree clean
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ ls
```

file1.txt

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ mv file1.txt main.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (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:    file1.txt
```

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

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

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file1.txt
```

```
g
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add main.js
warning: in the working copy of 'main.js', LF will be replaced by CRLF the next time
Git touches it
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    file1.txt -> main.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git mv main.js file1.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    file1.txt -> file1.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Refactor code"
[master 7aa9658] Refactor code
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename file1.txt => file1.js (100%)
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ mkdir logs
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo hello >> logs/dev.log
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logs/
```

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

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo logs/ > .gitignore
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ code .gitignore
```

```

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next
time Git touches it

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Add gitignore"
[master 3923062] Add gitignore
1 file changed, 3 insertions(+)
create mode 100644 .gitignore

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ mkdir bin

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo hello > bin/app.bin

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        bin/

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

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add .
warning: in the working copy of 'bin/app.bin', LF will be replaced by CRLF the next
time Git touches it

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Add bin."
[master 6191d28] Add bin.
1 file changed, 1 insertion(+)
create mode 100644 bin/app.bin

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ code .gitignore

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (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")

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next
time Git touches it

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git commit -m "Include bin/ in gitignore."
[master 92c2bb0] Include bin/ in gitignore.
1 file changed, 1 insertion(+)

De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo helloworld > bin/aap.bin

```



```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ echo helloworld > bin/app.bin
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (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:   bin/app.bin
```

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

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-files
.gitignore
bin/app.bin
file1.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm -h
usage: git rm [-f | --force] [-n] [-r] [--cached] [--ignore-unmatch]
              [--quiet] [--pathspec-from-file=<file>] [--pathspec-file-nul]]
              [--] [<pathspec>...]
```

-n, --dry-run	dry run
-q, --quiet	do not list removed files
--cached	only remove from the index
-f, --force	override the up-to-date check
-r	allow recursive removal
--ignore-unmatch	exit with a zero status even if nothing matched
--sparse	allow updating entries outside of the sparse-checkout cone
--pathspec-from-file <file>	read pathspec from file
--pathspec-file-nul	with --pathspec-from-file, pathspec elements are separated with NUL character

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm --cached bin/
fatal: not removing 'bin/' recursively without -r
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git rm -h
usage: git rm [-f | --force] [-n] [-r] [--cached] [--ignore-unmatch]
              [--quiet] [--pathspec-from-file=<file>] [--pathspec-file-nul]]
              [--] [<pathspec>...]
```

-n, --dry-run	dry run
-q, --quiet	do not list removed files
--cached	only remove from the index
-f, --force	override the up-to-date check
-r	allow recursive removal
--ignore-unmatch	exit with a zero status even if nothing matched
--sparse	allow updating entries outside of the sparse-checkout cone
--pathspec-from-file <file>	read pathspec from file
--pathspec-file-nul	with --pathspec-from-file, pathspec elements are separated with NUL character

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
```

```
$ git rm --cached -r bin/  
rm 'bin/app.bin'
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git ls-files  
.gitignore  
file1.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    deleted:   bin/app.bin
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git commit -m "Remoe the bin directory that was accidentally committed."  
[master 4c4678e] Remoe the bin directory that was accidentally committed.  
 1 file changed, 1 deletion(-)  
delete mode 100644 bin/app.bin
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ echo test > bin/app.bin
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git status  
On branch master  
nothing to commit, working tree clean
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git status -s
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ echo sky >> file1.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ echo sky > file2.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (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:   file1.js
```

```
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    file2.js
```

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

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git status -s  
  M file1.js  
  ?? file2.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git add file1.js  
warning: in the working copy of 'file1.js', LF will be replaced by CRLF the next time  
Git touches it
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ git status -s  
  M file1.js  
  ?? file2.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)  
$ echo ocean >> file1.js
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
```

```

$ git status -s
MM file1.js
?? file2.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file1.js
warning: in the working copy of 'file1.js', LF will be replaced by CRLF the next time
Git touches it

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status -s
M file1.js
?? file2.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git add file2.js
warning: in the working copy of 'file2.js', LF will be replaced by CRLF the next time
Git touches it

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status -s
M file1.js
A file2.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git diff --staged
diff --git a/file1.js b/file1.js
index badfb70..47c3216 100644
--- a/file1.js
+++ b/file1.js
@@ -1,3 +1,5 @@
hello
world
test
+sky
+ocean
diff --git a/file2.js b/file2.js
new file mode 100644
index 0000000..f5e95e7
--- /dev/null
+++ b/file2.js
@@ -0,0 +1 @@
+sky

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git diff

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status -s
M file1.js
A file2.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ code file1.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git status -s
MM file1.js
A file2.js

Dell@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git diff
warning: in the working copy of 'file1.js', LF will be replaced by CRLF the next time
Git touches it
diff --git a/file1.js b/file1.js
index 47c3216..1646bc2 100644
--- a/file1.js
+++ b/file1.js
@@ -1,4 +1,4 @@
-hello
+hello BWFS
world

```

test
sky

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ git log
commit 4c4678ec865de19320d61614a5342e1d031f7adf (HEAD -> master)
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:14:29 2023 +0500
```

Remoe the bin directory that was accidentally committed.

```
commit 92c2bb011abaf97c03930a25bb2d99129a0bfb3d
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:10:45 2023 +0500
```

Include bin/ in gitignore.

```
commit 6191d28295d5148f3022f548fed892d372d7bea5
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:08:21 2023 +0500
```

Add bin.

```
commit 39230626bb2f5c625de9e727be21c6cd27b8ec4e
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:05:58 2023 +0500
```

Add gitignore

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ git log --oneline
4c4678e (HEAD -> master) Remoe the bin directory that was accidentally committed.
92c2bb0 Include bin/ in gitignore.
6191d28 Add bin.
3923062 Add gitignore
7aa9658 Refactor code
9f577ec Remove unused code.
b994e3f Fix the bug.
e1825e3 Initial commit.
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ git log --oneline --reverse
e1825e3 Initial commit.
b994e3f Fix the bug.
9f577ec Remove unused code.
7aa9658 Refactor code
3923062 Add gitignore
6191d28 Add bin.
92c2bb0 Include bin/ in gitignore.
4c4678e (HEAD -> master) Remoe the bin directory that was accidentally committed.
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ git show ^C
```

```
De1l@DESKTOP-L7MC02J MINGW64 ~/DL-BWF (master)
$ git show 92c2bb0
commit 92c2bb011abaf97c03930a25bb2d99129a0bfb3d
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:10:45 2023 +0500
```

Include bin/ in gitignore.

```
diff --git a/.gitignore b/.gitignore
index b35834d..de80a7b 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,3 +1,4 @@
 logs/
 main.log
 *.log
+bin/
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git show HEAD~2
commit 6191d28295d5148f3022f548fed892d372d7bea5
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:08:21 2023 +0500
```

Add bin.

```
diff --git a/bin/app.bin b/bin/app.bin
new file mode 100644
index 0000000..ce01362
--- /dev/null
+++ b/bin/app.bin
@@ -0,0 +1 @@
+hello
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git show HEAD~1:.gitignore
logs/
main.log
*.log
bin/
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git show HEAD
commit 4c4678ec865de19320d61614a5342e1d031f7adf (HEAD -> master)
Author: Munaza Ashraf <munazaa85@gmail.com>
Date: Thu Mar 16 20:14:29 2023 +0500
```

Remoe the bin directory that was accidentally committed.

```
diff --git a/bin/app.bin b/bin/app.bin
deleted file mode 100644
index ce01362..0000000
--- a/bin/app.bin
+++ /dev/null
@@ -1 +0,0 @@
-hello
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-tree
usage: git ls-tree [<options>] <tree-ish> [<path>...]

    -d                only show trees
    -r                recurse into subtrees
    -t                show trees when recursing
    -z                terminate entries with NUL byte
    -l, --long        include object size
    --name-only       list only filenames
    --name-status     list only filenames
    --object-only     list only objects
    --full-name       use full path names
    --full-tree       list entire tree; not just current directory (implies --
full-name)
    --format <format> format to use for the output
    --abbrev[=<n>]    use <n> digits to display object names
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git ls-tree HEAD~1
100644 blob de80a7b9265aa3e11b1a8804a3c62a9492a0e23d    .gitignore
040000 tree 64629cd51ef4a65a9d9cb9e656e1f46e07e1357f    bin
100644 blob badfb70fd8b1725682b26674f7b2882e94078579    file1.js
```

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$ git show 64629
tree 64629
```

app.bin

```
De1l@DESKTOP-L7MCO2J MINGW64 ~/DL-BWF (master)
$
```