

# GIT EXERCISES

## 1.master

### commands used.

\$ git start master

\$ git push

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the master exercise. Hold on...
Exercise: master
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/master/wja

Next task: commit-one-file
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

Exercise: Push a commit you have made

You have passed this exercise!

`git start master` Copy

**Instructions**  
The first exercise is to push a commit that is created when you run the `git start` command.  
Just try `git verify` after you have initialized the exercises and be proud of passing the first one :-)

**The easiest solution**  
`git verify` Copy

master ✓

commit-one-file

commit-one-file-staged

ignore-them

chase-branch

merge-conflict

save-your-work

change-branch-history

remove-ignored

case-sensitive-filename

fix-typo

forge-date

## 2.commit one file

**Aim :** commit only one of the two files present.

### Commands used

\$ git start commit-one-file

\$ git add A.txt

\$ git commit -m "Add A.txt"

\$ git verify

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the commit-one-file exercise. Hold on...
Exercise: commit-one-file
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/commit-one-file/wja

Next task: commit-one-file-staged
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

## Exercise: Commit one file

You have passed this exercise!

```
git start commit-one-file
```

✓ Copied

### Instructions

There are two files created in the root project directory - A.txt and B.txt .

The goal is to commit only one of them.

NOTE: Remember that you can submit your solutions with `git verify` command instead of `git push` .

- master ✓
- commit-one-file ✓
- commit-one-file-staged
- ignore-them
- chase-branch
- merge-conflict
- save-your-work
- change-branch-history
- remove-ignored
- case-sensitive-filename

### 3.commit one file staged

**Aim:** commit only one of the two files staged for commit

#### commands used

```
$ git start commit-one-file-staged
```

```
$ git status
```

```
$ git reset B.txt
```

Then there are two files staged .So we have to reset one among them and to commit the other one.

```
$ git commit -m "commit only A"
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the commit-one-file-staged exercise. Hold on...
Exercise: commit-one-file-staged
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/commit-one-file-staged/wja

Next task: ignore-them
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

## Exercise: Commit one file of two currently staged

You have passed this exercise!

```
git start commit-one-file-staged
```

Copy

### Instructions

There are two files created in the root project directory - A.txt and B.txt . They are both added to the staging area.

The goal is to commit only one of them.

### 4. Ignore unwanted files

**Aim:** ignore the files with given file-extensions

#### commands used

```
$ git start ignore-them
```

```
$ nano .gitignore
```

```
inside nano.gitignore
```

```
*.o
*.exe
*.jar
libraries/
$ git add .gitignore
$ git commit -m "Add .gitignore"
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git ve
Verifying the ignore-them exercise. Hold on...
Exercise: ignore-them
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/ignore-them/wja

Next task: chase-branch
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

### Exercise: Ignore unwanted files

You have passed this exercise!

git start ignore-them

Copy

#### Instructions

It is often good idea to tell Git which files it should track and which it should not. Developers almost always do

master	✓
commit-one-file	✓
commit-one-file-staged	✓
ignore-them	✓
chase-branch	
merge-conflict	
save-your-work	
change-branch-history	

## 5. Chase branch that escaped

**Aim:** make chase-branch to point to the same commit as the escaped branch.

### Commands used

```
$ git start chase-branch
$ git merge escaped
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the chase-branch exercise. Hold on...
Exercise: chase-branch
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/chase-branch/wja

Next task: merge-conflict
In order to start, execute: git start next
```

### Exercise: Chase branch that escaped

You have passed this exercise!

```
git start chase-branch
```

Copy

Instructions

- master ✓
- commit-one-file ✓
- commit-one-file-staged ✓
- ignore-them ✓
- chase-branch ✓
- merge-conflict
- save-your-work

## 6.Resolve a merge conflict

**Aim :** merge the another-piece-of-work into your current branch. This will cause a merge conflict which you have to resolve.

### Commands used

```
$ git start merge-conflict
$ git merge another-piece-of-work
$ nano equation.txt
$ git add equation.txt
$ git commit -m "Add equation.txt"
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ nano equation.txt
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git add equation.txt
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git commit -m "try"
[merge-conflict c040af8] try
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the merge-conflict exercise. Hold on...
Exercise: merge-conflict
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/merge-conflict/wja
```

### Exercise: Resolve a merge conflict

You have passed this exercise!

```
git start merge-conflict
```

Copy

- master ✓
- commit-one-file ✓
- commit-one-file-staged ✓
- ignore-them ✓
- chase-branch ✓
- merge-conflict ✓

## 7.Saving your work

**Aim:**commit work after bugfix.

### Commands used

```
$ git start save-your-work
$ git stash
$ nano bug.txt
$ git add bug.txt
$ git commit -m "Fixed bug"
```

```
$ git stash pop
$ git commit -am "done"
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the save-your-work exercise. Hold on...
Exercise: save-your-work
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/save-your-work/wja

Next task: change-branch-history
In order to start, execute: git start next
```

### Exercise: Saving your work

You have passed this exercise!

```
git start save-your-work
```

Copy

#### Instructions

You are working hard on a regular issue while your boss comes in and wants you to fix a bug. State of your

master	✓
commit-one-file	✓
commit-one-file-staged	✓
ignore-them	✓
chase-branch	✓
merge-conflict	✓
save-your-work	✓
change-branch-history	

## 8.Change branch history

**Aim :** *rebase* our current branch change-branch-history

### Commands used

```
$git rebase hot-bugfix
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the change-branch-history exercise. Hold on...
Exercise: change-branch-history
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/change-branch-history/wja

Next task: remove-ignored
In order to start, execute: git start next
```

### Exercise: Change branch history

You have passed this exercise!

```
git start change-branch-history
```

Copy

#### Instructions

master	✓
commit-one-file	✓
commit-one-file-staged	✓
ignore-them	✓
chase-branch	✓
merge-conflict	✓
save-your-work	✓

## 9.Remove ignored file

**Aim :** File ignored.txt is ignored by rule in .gitignore but is tracked because it had been added before the ignoring rule was introduced . Remove it so changes in ignored.txt file are not tracked anymore.

### Commands used

```
$ git rm ignored.txt  
$ git commit -m "removed"  
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify  
Verifying the remove-ignored exercise. Hold on...  
Exercise: remove-ignored  
Status: PASSED  
You can see the easiest known solution and further info at:  
https://gitexercises.fracz.com/e/remove-ignored/wja  
  
Next task: case-sensitive-filename  
In order to start, execute: git start next
```

### Exercise: Remove ignored file

You have passed this exercise!

git start remove-ignored

Copy

master  
commit-  
commit-  
ignore-th  
chase-b  
merge-c

## 10.Change a letter case in the filename of an already tracked file

**Aim :** You have committed a File.txt but then you realized the filename should be all lowercase: file.txt. Change the filename.

### Commands used

```
$ git mv File.txt file.txt  
$ git commit -m "renamed"  
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify  
Verifying the case-sensitive-filename exercise. Hold on...  
Exercise: case-sensitive-filename  
Status: PASSED  
You can see the easiest known solution and further info at:  
https://gitexercises.fracz.com/e/case-sensitive-filename/wja  
  
Next task: fix-typo  
In order to start, execute: git start next
```

## Exercise: Change a letter case in the filename of an already tracked file

You have passed this exercise!

```
git start case-sensitive-filename
```

✓ Copied

### 11. Fix typographic mistake in the last commit

**Aim :** You have committed file.txt but you realized you made a typo - you wrote wordl instead of world. Edit previous commit so no one would realize you haven't checked the file before committing it.

#### Commands used

```
$ nano file.txt  
$ git add file.txt  
$ git commit --amend  
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify  
Verifying the fix-typo exercise. Hold on...  
Exercise: fix-typo  
Status: PASSED  
You can see the easiest known solution and further info at:  
https://gitexercises.fracz.com/e/fix-typo/wja  
  
Next task: forge-date  
In order to start, execute: git start next
```

## Exercise: Fix typographic mistake in the last commit

You have passed this exercise!

```
git start fix-typo
```

✓ Copied

### 12. Forge the commit's date

**Aim :** Change the date of the last commit. Don't be modest - make it look like it was committed in 1987

#### Commands used

```
$ git commit --amend --no-edit --date="1987-01-01"  
$ git verify
```



```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the forge-date exercise. Hold on...
Exercise: forge-date
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/forge-date/wja

Next task: fix-old-typo
In order to start, execute: git start next
```

### Exercise: Forge the commit's date

You have passed this exercise!

git start forge-date

✓ Copied

## 13. Fix typographic mistake in old commit

**Aim :** Fix the typographic error by amending commit in history.

### Commands used

```
$ git rebase -i HEAD~2 // change the pick to edit on Add hello world
$ nano file.txt // change wordl to world
$ git add file.txt
$ git rebase --continue //here a conflict appears , open file.txt again and remove conflict markers.
$ nano file.txt
$ git add file.txt
$ git rebase --continue
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the fix-old-typo exercise. Hold on...
Exercise: fix-old-typo
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/fix-old-typo/wja

Next task: commit-parts
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

### Exercise: Fix typographic mistake in old commit

You have passed this exercise!

git start fix-old-typo

✓ Copied



## 14. Find a commit that has been lost

**Aim :** find the first version of the commit in the repository.

### Commands used

```
$git reflog
```

```
$ git reset --hard HEAD@{1}
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the commit-lost exercise. Hold on...
Exercise: commit-lost
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/commit-lost/wja

Next task: fix-old-typo
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

Exercise: Find a commit that has been lost

You have passed this exercise!

git start commit-lost

✓ Copied

## 15. Split the last commit

**Aim :** You have committed both first.txt and second.txt as one commit. However, you made a mistake. You intended to commit first.txt in the first commit and the second.txt in the second one.

### Commands used

```
$ git reset
```

```
$ git add first.txt
```

```
$ git commit -m "first"
```

```
$ git add second.txt
```

```
$ git commit -m "second"
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the split-commit exercise. Hold on...
Exercise: split-commit
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/split-commit/wja

Next task: fix-old-typo
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

Exercise: Split the last commit

You have passed this exercise!

git start split-commit

✓ Copied

## 16. Too many commits

**Aim :** to make only one commit that contains changes made in the last two commits.

### Commands used

```
$ git rebase -i HEAD~2 //then make edit to squash for the second commit.
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify too-many-commits
Verifying the too-many-commits exercise. Hold on...
Exercise: too-many-commits
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/too-many-commits/wja

Next task: fix-old-typo
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

## Exercise: Too many commits

You have passed this exercise!

```
git start too-many-commits
```

Copy

## 17. Make the file executable by default

**Aim :** You have created a simple bash script in `script.sh`. However, when you check it out on Unix, it does not have required execute permissions so you can't launch it with `./script.sh` without performing `chmod +x script.sh` beforehand.

### Commands used

```
$ git update-index --chmod=+x script.sh
```

```
$ git commit -m "changed index"
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the executable exercise. Hold on...
Exercise: executable
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/executable/wja

Next task: fix-old-typo
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

## Exercise: Make the file executable by default

You have passed this exercise!

```
git start executable
```

Copy

### 18.Commit part of work

**Aim:** You are working on an issue for a long time and you noticed you have done too much. You want your work to be committed in two separate commits instead of one.

#### Commands used

```
$ git start commit-parts
$ git add -p    // select the s(split) option and select lines with phrase "task1"
$ git commit -m "part1"
$ git commit -m "part2"    // commit the remaining
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git commit -m "part2"
[commit-parts 6e1fbce] part2
1 file changed, 3 insertions(+), 1 deletion(-)
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the commit-parts exercise. Hold on...
Exercise: commit-parts
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/commit-parts/wja
Next task: find-swearwords
In order to start, execute: git start next
```

## Exercise: Commit part of work

You have passed this exercise!

```
git start commit-parts
```

✓ Copied

### 19.Pick your features

**Aim :** You have implemented three different features of a program in three different local topic branches.You want to have all these features as single commits in pick-your-features branch.

#### Commands used

```
$ git cherry-pick feature-a
$ git cherry-pick feature-b
$ git cherry-pick feature-c
solve the conflict in program.txt
```

```
$ git add program.txt
$ git cherry-pick --continue
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the pick-your-features exercise. Hold on...
Exercise: pick-your-features
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/pick-your-features/wja

Next task: fix-old-typo
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

## Exercise: Pick your features

You have passed this exercise!

```
git start pick-your-features
```

✓ Copied

## 20. Rebase complex

**Aim :** Get all commits that are not in issue-555 and place them onto your -master branch.

### Commads used

```
$ git rebase --onto your-master 8c40bb2 rebase-complex
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the rebase-complex exercise. Hold on...
Exercise: rebase-complex
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/rebase-complex/wja

Next task: commit-parts
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
```

## Exercise: Rebase complex

You have passed this exercise!

```
git start rebase-complex
```

📋 Copy

## 21.Change order of commits

**Aim :** You have committed two changes but you don't like the order in which they appear in the history. Switch them.

### Commands used

```
$ git rebase -i HEAD~2 //Then move the second commit up
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the invalid-order exercise. Hold on...
Exercise: invalid-order
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/invalid-order/wja

Next task: commit-parts
In order to start, execute: git start next

See your progress and instructions at:
https://gitexercises.fracz.com/c/wja
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$
```

## Exercise: Change order of commits

You have passed this exercise!

```
git start invalid-order
```

Copy

## 22.Find commits that introduced swearwords

**Aim :** You noticed that your project contains swearwords. You don't want to remove them in the next commit as your boss might someday find out that they were present in the codebase for a while.

Find all commits that add *shit* word to either words.txt or list.txt and change them so they introduce word *flower* instead.

### Commands used

```
$ git log -S shit
```

```
$ git rebase -i HEAD~105
```

```
$ git log -p -1
```

```
$ nano list.txt
```

```
$ git commit -a --amend
```

```
$ git rebase --continue // do this for two more times .
```

```
$ git verify
```

```
prithewjraj@prithewjraj-HP-Laptop-15s-fq5xxx:~/exercises$ git verify
Verifying the find-swearwords exercise. Hold on...
Exercise: find-swearwords
Status: PASSED
You can see the easiest known solution and further info at:
https://gitexercises.fracz.com/e/find-swearwords/wja

Next task: find-bug
In order to start, execute: git start next
```

## Exercise: Find commits that introduced swearwords

You have passed this exercise!

git start find-swearwords

✓ Copied

### 23. Find commit that has introduced bug

**Aim :** task is to find the first commit that introduces the *jackass* word and push it for verification.

#### Commands used

```
$ git start find-bug
```

```
$ git bisect start
```

```
$ git bisect bad HEAD
```

```
$ git bisect good 1.0 //The problem states that version tagged as 1.0 didn't have the bug.
Therefore, we'll categorize the head as bad and this commit as good.
```

```
$ git bisect run sh -c "openssl enc -base64 -A -d < home-screen-text.txt | grep -v jackass"
```

```
$ git bisect reset //commit #78 identified as the first bad commit.
```

```
$ git push origin 06eb812:find-bug
```

```
Compressing objects: 100% (229/229), done.
Writing objects: 100% (237/237), 30.43 KiB | 2.54 MiB/s, done.
Total 237 (delta 0), reused 158 (delta 0), pack-reused 0
remote: (
remote: *****
remote: Exercise: find-bug
remote: Status: PASSED
remote: You can see the easiest known solution and further info at:
remote: https://gitexercises.fracz.com/e/find-bug/wja
remote:
remote: Congratulations! You have done all exercises!
```

## Exercise: Find commit that has introduced bug

You have passed this exercise!

git start find-bug

📋 Copy

## Personal experience

Working on the git exercises was a very good experience . The initial levels were simple and easy to follow , but as the levels increased , the tasks became more challenging and time consuming . Overall it was a valuable learning journey.

