

Start Work

```
$ git fetch origin
$ git switch -c <branch name> origin/main
$ git branch -D main master
```

Parallel Development

```
$ git fetch origin
Grab the latest from remote
$ git switch -c <branch> origin/main
Create a branch off the mainline
$ git rebase origin/main
Update branch to mainline
$ git worktree add <newpath> origin/main -b <hotpatch>
Create a new repo copy to develop a hotpatch
```

Move Work

```
$ git cherry-pick <commit>
$ git rebase <branch name>
$ git rebase <commit>
Find <commit> with $ gitk
```

Test Committed

```
$ git stash --include-untracked
$ git stash pop
```

Collaborate

```
$ git fetch origin
$ git rebase origin/main
$ git rebase --continue
$ git push [--force] origin <branch>
Include --force on personal branch
$ git push origin --delete <branch>
Delete Remote Branch
```

Work in progress

```
$ git commit --fixup HEAD
$ git config --global alias.fixup 'commit --fixup'
$ git commit --fixup <commit>
$ git commit --fixup :/<message>
$ git rebase -i <commit>~1
$ git config --global rebase.autosquash true
```

Fix Work

Fix Commit Message

```
$ git rebase -i origin/main reword
```

Important Commit First

```
$ git rebase -i origin/main Reorder lines
```

Combine Commits

```
$ git rebase -i origin/main fixup
$ git rebase -i origin/main squash
```

Break Apart Work

```
$ git rebase -i origin/main edit
```

```
$ git gui
```

Use the "Unstage Lines" feature

The icon operates the whole file

Save Work

```
$ git add <file>
$ git commit
$ git stash push -m <message>
$ git stash apply stash^/<message>
```

Clean Up

```
$ git fetch origin --prune
$ git switch --detach origin/main
$ git branch --merged
$ git branch --delete <branch1> <branch2>
$ git push origin --delete <branch>
```

<commit>

```
Hash      :/msg
Branch    <branch>:/msg
origin/branch
```

Branch or Remote

```
origin/main  <branch>
origin        <remote>
$ git push origin  push <remote> <branch>
$ git fetch origin  fetch <remote>
$ git rebase      rebase <branch>
origin/main
```

Cancel That

```
$ git rebase --abort
$ git merge --abort
$ git cherry-pick --abort
$ git reset --hard <commit>
```

If the operation completed, find your starting point:

```
$ git reflog
```

Don't Do

```
$ git reset --hard
$ git clean --force
$ git switch <branch> --force
```

These types of commands will destroy data, not retrievable.



By **Jesse K Phillips**
(JesseKPhillips)

cheatography.com/jessekphillips/

Published 8th April, 2021.

Last updated 4th May, 2021.

Page 1 of 1.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopod.com>