

Git Commands: <https://www.atlassian.com/git/glossary>

Config:

```
git config --global user.name "Your Name"  
git config --global user.email your.email@example.com  
git config --list
```

Add – to add files to stage area.

```
git add <file>  
git add <directory>  
git add .
```

→ to remove file from stage area

```
git restore --staged <exact_file_path_and_name>. (U can use remove command also -- git rm  
--cached src/remove.java.)
```

after adding to stage area but you don't want some file/folder to commit then use this command to remove from staged area

Commit: -- to save changes in local repo

```
git commit
```

it will open text editor to take long commit message then close it

```
git commit -m "meaning full commit message."
```

```
git commit --amend
```

This will open new text editor to take commit message

```
git commit --amend -m "commit message"
```

Recommend: git commit --amend -m "sorry forgot to add this changes in previous commit so replace previous commit instead of creating new commit"

Not recommend shortcut: git commit -am "added remaining files"

```
git commit -m "COMMIT MESSAGE"  
git add forgotten_file  
git commit --amend -m "Updated message 4 all files"
```

Branch:

`git branch -list`

List all the branches in your local repository

`git branch -a`

List all remote branches.

`git branch <branch_name>` – default source branch_name is master

`git branch <branch_name> <source_or_base_branch_name>`

Create a new branch called `<branch>` in local repo. This does not check out the new branch.

`git branch -d <branch>`

Delete the specified branch. This is a “safe” operation in that Git prevents you from deleting the branch if it has unmerged changes.

`git branch -D <branch>`

Force delete the specified branch, even if it has unmerged changes. This is the command to use if you want to permanently throw away all of the commits associated with a particular line of development.

`git branch -m <branch>`

Rename the current branch to `<branch>`

`git checkout <new-branch>`

enter into specific branch from current branch

`git checkout -b <new-branch>`

create new branch and entered into created branch.

`git checkout <commit-id>`

checkout to specific git commit.

Checkout to remote branch:

```
gitpod /workspace/IntelliJ-IDE-Overview (main) $ git branch -a
  login
  logout
* main
  signup
  remotes/origin/main
  remotes/origin/signup
gitpod /workspace/IntelliJ-IDE-Overview (main) $ git checkout remotes/origin/signup
```

```
git checkout -b <new-branch-name> <existing-branch-name>
```

creating new branch from existing branch

```
diff -d Vs -D
```

if branch already merged then use

```
git branch -d <branch-name>
```

if branch not merged but still if you want delete branch then use below command

```
git branch -D <branch-name>
```

Remote:

```
git push origin --delete <remote_branch_name>
```

delete remote branch after merge to main or if you don't want that specific branch changes then delete that branch.

```
git clone <repo link>
```

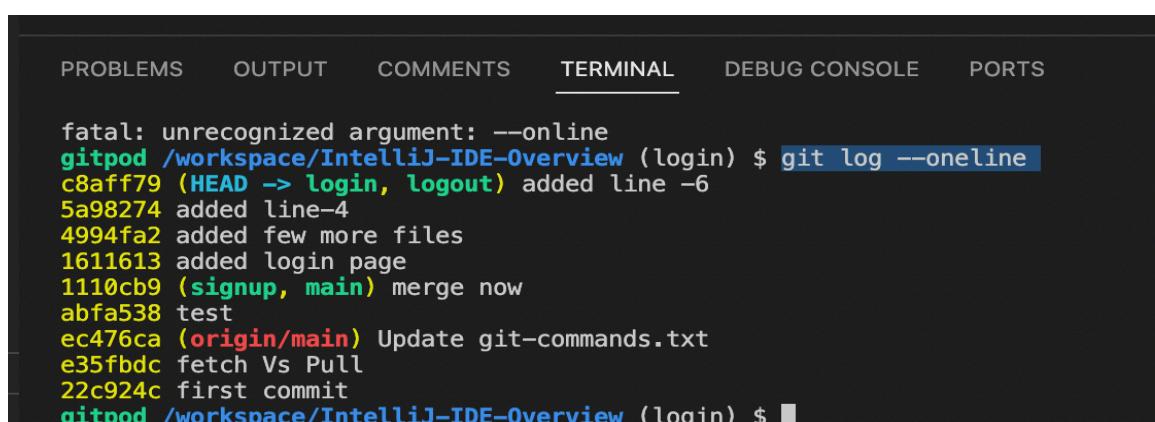
to get remote repository into local machine

History:

```
git log
```

it will show all commits in current branch.

```
git log --oneline
```



The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, COMMENTS, TERMINAL (which is underlined), DEBUG CONSOLE, and PORTS. The terminal content displays a list of Git commit messages in a single-line format (oneline). The commits are as follows:

```
fatal: unrecognized argument: --online
gitpod /workspace/IntelliJ-IDE-Overview (login) $ git log --oneline
c8aff79 (HEAD -> login, logout) added line -6
5a98274 added line-4
4994fa2 added few more files
1611613 added login page
1110cb9 (signup, main) merge now
abfa538 test
ec476ca (origin/main) Update git-commands.txt
e35fbdc fetch Vs Pull
22c924c first commit
gitpod /workspace/IntelliJ-IDE-Overview (login) $ █
```

```
git log --branches=*
```

it will show all commits on all branches

Undo

```
git revert <commit_id>
```

before revert

```
HelloJava > src > Hello > main
HelloJava ~/IdeaProjects/HelloJava
Project Commit Bookmarks Structure
HelloJava ~/IdeaProjects/HelloJava
  > .idea
  > out
  > src
    Hello
    Main
    HelloJava.iml
  > External Libraries
  > Scratches and Consoles

HelloJava.iml x Hello.java x
no usages
public class Hello {
  no usages
  public static void main(String args[]){
    System.out.println("Hello Maldanna");
    System.out.println("line-1");
  }
}

Terminal: Local x + ▾
maldann.gattannagari@MREMB58E7248 HelloJava % git commit -m "line-1"
[master bd0fc5c] line-1
  1 file changed, 1 insertion(+)
maldann.gattannagari@MREMB58E7248 HelloJava % git log --oneline
bd0fc5c (HEAD -> master) line-1
8f771f4 first commit with hello maldanna
maldann.gattannagari@MREMB58E7248 HelloJava %
```

After revert:

```
HelloJava – Hello.java
HelloJava > src > Hello > main
HelloJava ~/IdeaProjects/HelloJava
Project Commit Bookmarks Structure
HelloJava ~/IdeaProjects/HelloJava
  > .idea
  > out
  > src
    Hello
    Main
    HelloJava.iml
  > External Libraries
  > Scratches and Consoles

HelloJava.iml x Hello.java x
no usages
public class Hello {
  no usages
  public static void main(String args[]){
    System.out.println("Hello Maldanna");
  }
}

Terminal: Local x + ▾
maldann.gattannagari@MREMB58E7248 HelloJava % git log --oneline
a029ca6 (HEAD -> master) Revert "line-1"
bd0fc5c Line-1
8f771f4 first commit with hello maldanna
maldann.gattannagari@MREMB58E7248 HelloJava %
```

Link: <https://www.youtube.com/watch?v=hA3vYDyaZQ0>

If you want that changes again then revert on previous revert command

The screenshot shows the IntelliJ IDEA interface with a Java project named "HelloJava". The "src" directory contains files "Hello.java" and "Main.java". The "Hello.java" file has the following code:

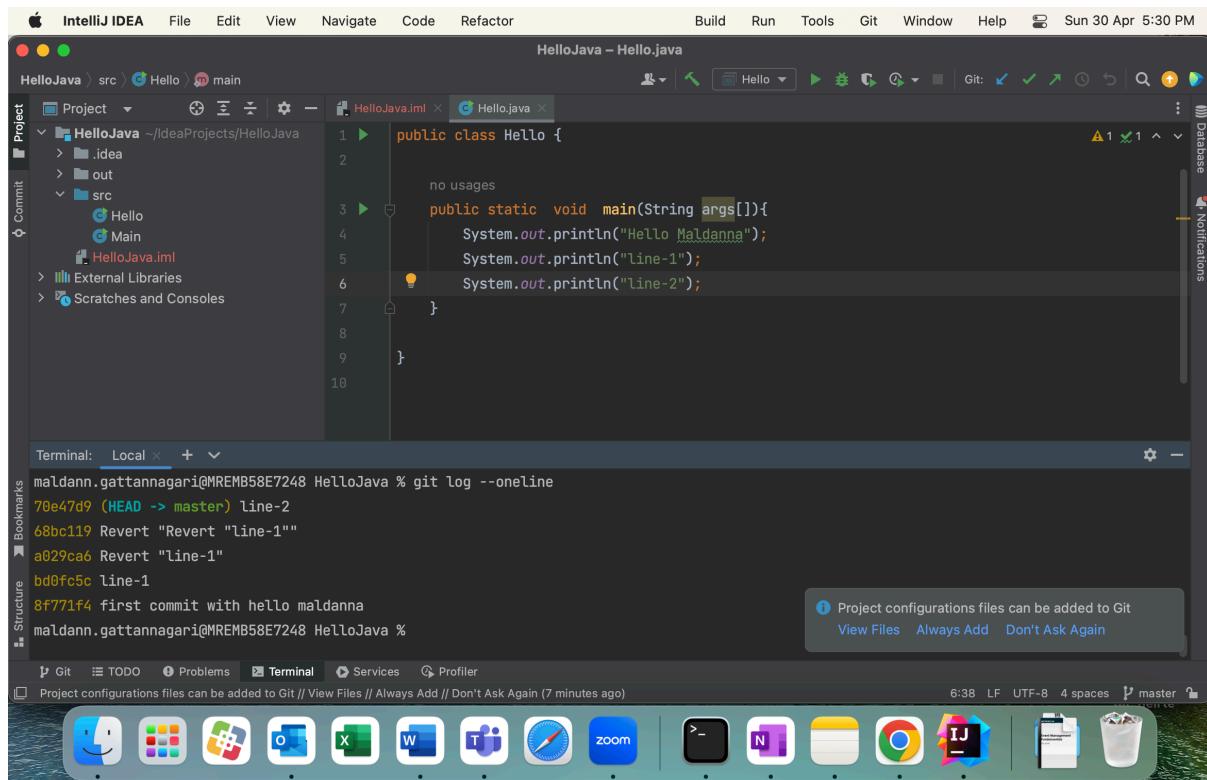
```
public class Hello {  
    public static void main(String args[]){  
        System.out.println("Hello Maldanna");  
        System.out.println("line-1");  
    }  
}
```

The "Terminal" window at the bottom shows the following git log output:

```
1 file changed, 1 insertion(+)  
maldann.gattannagari@MREMB58E7248 HelloJava % git log --oneline  
68bc119 (HEAD -> master) Revert "Revert "line-1""  
a029ca6 Revert "line-1"  
bd0fc5c line-1  
8f771f4 first commit with hello maldanna  
maldann.gattannagari@MREMB58E7248 HelloJava %
```

git reset <commit-id> (not recommend – use revert command for undo changes)

before run:



See All files in different area:



To see in current working directory -- Command – ls
To see all files in stage area -- Command – git ls-files

Remove file:

1. **git rm <file_name>.** --- if you want remove file from both working directory and staging area

2. git rm -r . – to remove all files from both areas
3. git rm --cached <file_name>. – to remove file from staging area (U can use unstage command also)
4. rm <file-name> -- to remove file from working directory

Complex Commands:

Merge

Merge login_feature(source) branch into master (target)

Step1: checkout to target branch. – git checkout master

Step2: git merge <source-branch> -- git merge login-feature

Rebase (similar to merge but small change): -

<https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>

Rebasing is the process of moving or combining a sequence of commits to a new base commit.

Merge lot of commits into single commit –

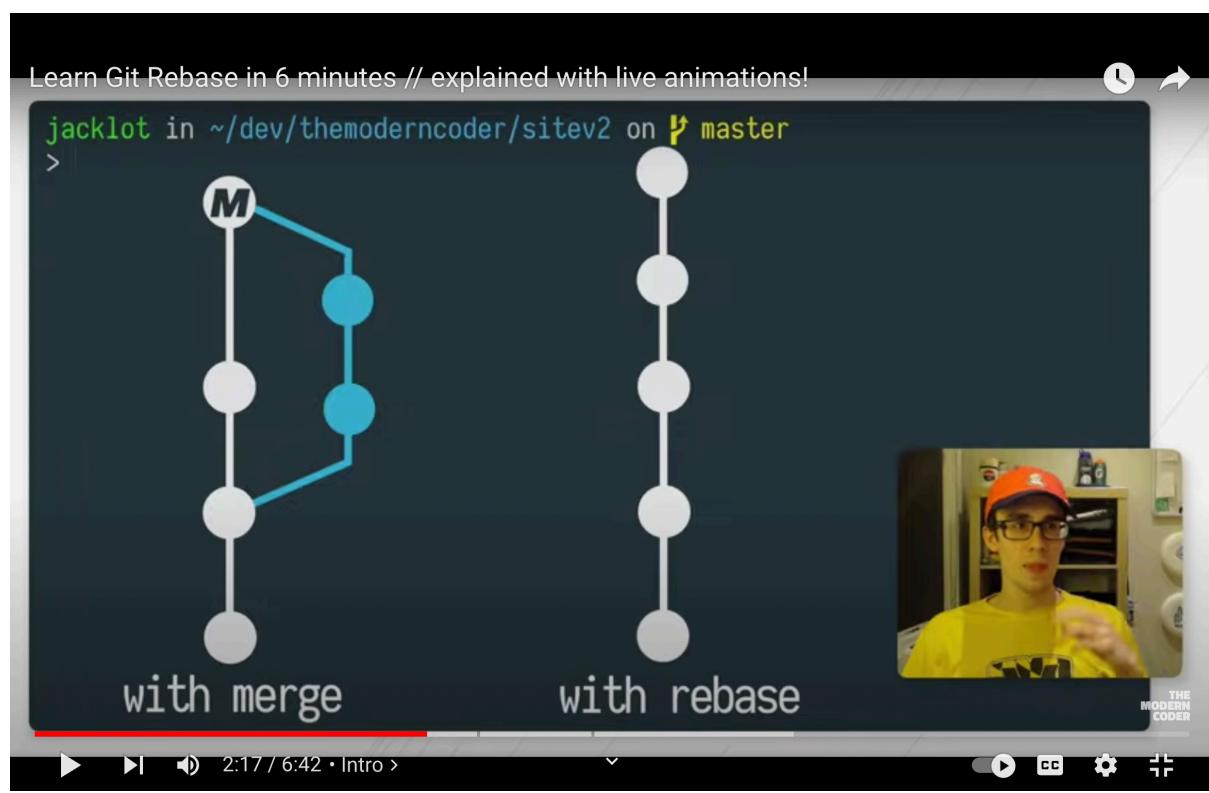
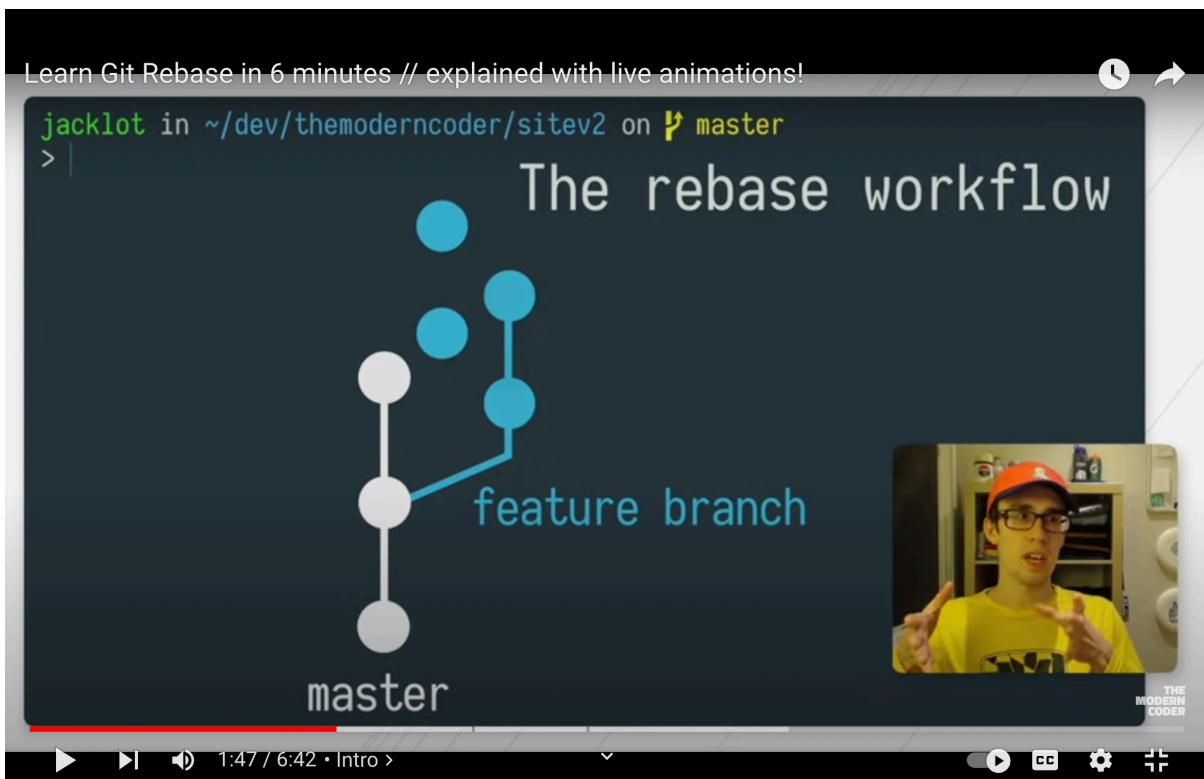
Step1: checkout to master

Step2: git rebase <branch-name>

Step3: resolve conflict

Step4: commit on master

Diff : <https://www.youtube.com/watch?v=f1wnYdLEpgI>



PULL (fetch + merge)

```
git pull <remote> <branch-name>
1. git clone
2. create login branch
3. checkout to login
4. make modification'
5. commit in login
6. checkout to main
7. pull from remote to make upto date
8. resolve merge conflicts.
9. Merge from login into main'
10. Resolve conflicts
11. Commit in main
12. Push to origin
```

git pull - pulling from remote main to local main.

pull into specific branch:

```
step1: checkout into specific branch
step2: git pull <remote> <branch-name>
```

presteps:

```
git remote -v
```

```
git remote
```

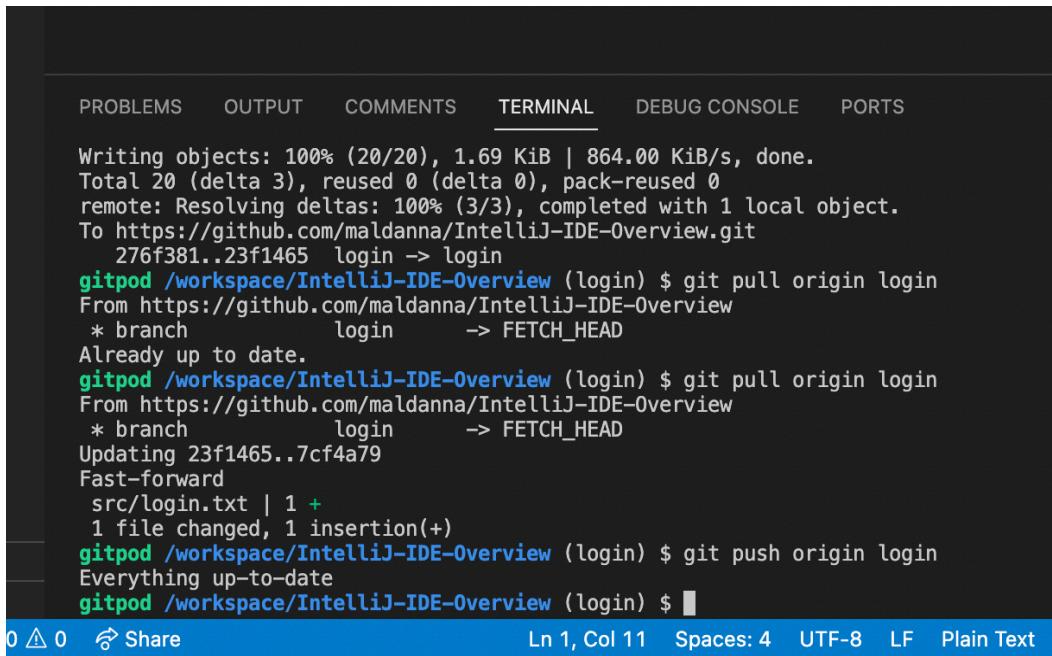
```
● gitpod /workspace/IntelliJ-IDE-Overview (login) $ git remote
origin
● gitpod /workspace/IntelliJ-IDE-Overview (login) $ git remote -v
origin  https://github.com/maldanna/IntelliJ-IDE-Overview.git (fetch)
origin  https://github.com/maldanna/IntelliJ-IDE-Overview.git (push)
○ gitpod /workspace/IntelliJ-IDE-Overview (login) $ []
```

Git pull from remote main to local branch.

```
1 file changed, 1 insertion(+)
⑧ gitpod /workspace/IntelliJ-IDE-Overview (login) $ git pull origin main
From https://github.com/maldanna/IntelliJ-IDE-Overview
 * branch           main      -> FETCH_HEAD
Auto-merging src/login.txt
CONFLICT (content): Merge conflict in src/login.txt
Automatic merge failed; fix conflicts and then commit the result.
○ gitpod /workspace/IntelliJ-IDE-Overview (login|MERGING) $ []
```

PUSH: git push <remote> <branch-name>

1. From local main to remote main
Step1: checkout to main
Step2: git pull.
Step3: resolve conflicts.
Step4: commit
Step5: git push
2. From local branch to remote branch
Step: checkout to that brach
Step2: do modifications
Step3: git pull from main or specific branch.
Step4: resolve conflicts.
Step5: push to specific branch
Like git push origin login



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT COMMENTS TERMINAL DEBUG CONSOLE PORTS

Writing objects: 100% (20/20), 1.69 KiB | 864.00 KiB/s, done.
Total 20 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/maldanna/IntelliJ-IDE-Overview.git
  276f381..23f1465  login -> login
gitpod /workspace/IntelliJ-IDE-Overview (login) $ git pull origin login
From https://github.com/maldanna/IntelliJ-IDE-Overview
 * branch            login      -> FETCH_HEAD
Already up to date.
gitpod /workspace/IntelliJ-IDE-Overview (login) $ git pull origin login
From https://github.com/maldanna/IntelliJ-IDE-Overview
 * branch            login      -> FETCH_HEAD
Updating 23f1465..7cf4a79
Fast-forward
 src/login.txt | 1 +
 1 file changed, 1 insertion(+)
gitpod /workspace/IntelliJ-IDE-Overview (login) $ git push origin login
Everything up-to-date
gitpod /workspace/IntelliJ-IDE-Overview (login) $ █
```

At the bottom of the terminal window, there is a status bar with the following information:

0 △ 0 Share Ln 1, Col 11 Spaces: 4 UTF-8 LF Plain Text

Fetch:

```
git fetch <remote> <branch-name>
```

step1: git fetch origin login → it will retrieve code from remote repo keep in ur local machine, but it won't show in

```
gitpod /workspace/IntelliJ-IDE-Overview (login) $ git diff login origin/login
diff --git a/src/login.txt b/src/login.txt
index e13315d..a54aac6 100644
--- a/src/login.txt
+++ b/src/login.txt
@@ -1,3 +1,4 @@
 login page
 test me
 test me2
+test4
gitpod /workspace/IntelliJ-IDE-Overview (login) $
```

△ 0 ⚡ Share Ln 1, Col 11 Spaces: 4 UTF-8 LF Plain Text