

GIT COMMANDS

1. Git setup

SET CONFIG VALUES

```
$ git config --global user.name "Corey Schafer"
$ git config --global user.email "CoreyMSchafer@gmail.com"
$ git config --list
```

2. Help

NEED HELP?

- git help <verb>
- git <verb> --help

```
$ git help config
$ git config --help
```

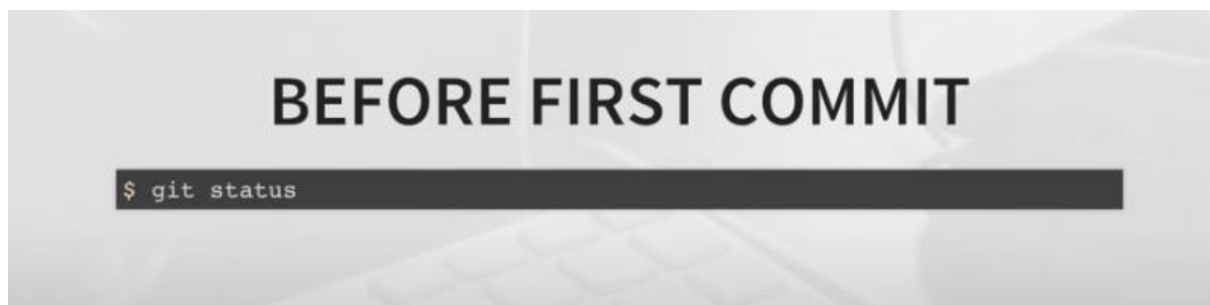
3. Initialize

INITIALIZE A REPOSITORY FROM EXISTING CODE

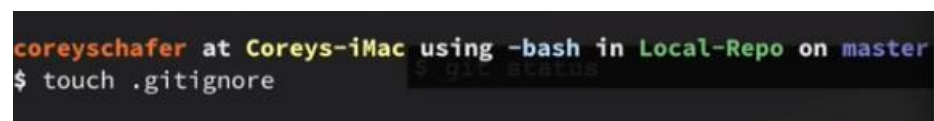
```
$ git init
```

rm -rf .git // to remove git init file.

4. Git status



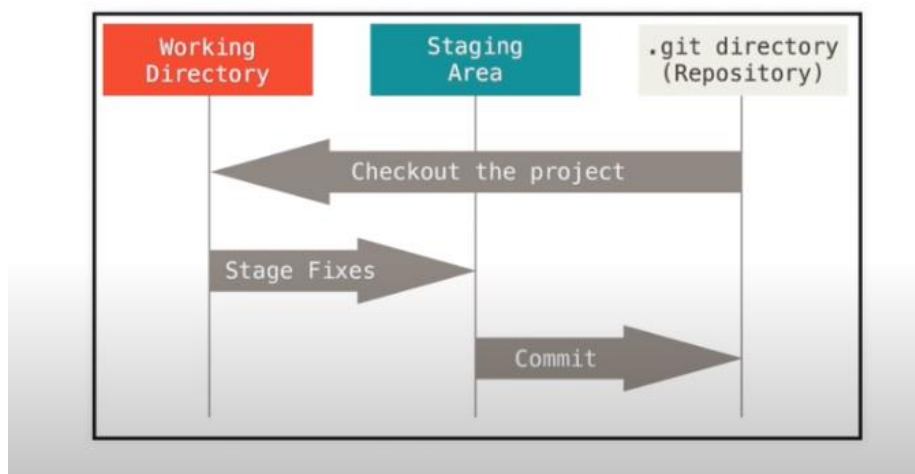
5. Creating git ignore file



6. Adding files in gitignore



WHERE ARE WE NOW?



7. Add files to staging area

ADD FILES TO STAGING AREA

```
$ git add -A  
$ git status
```

8. Removing files from staging area

REMOVE FILES FROM STAGING AREA

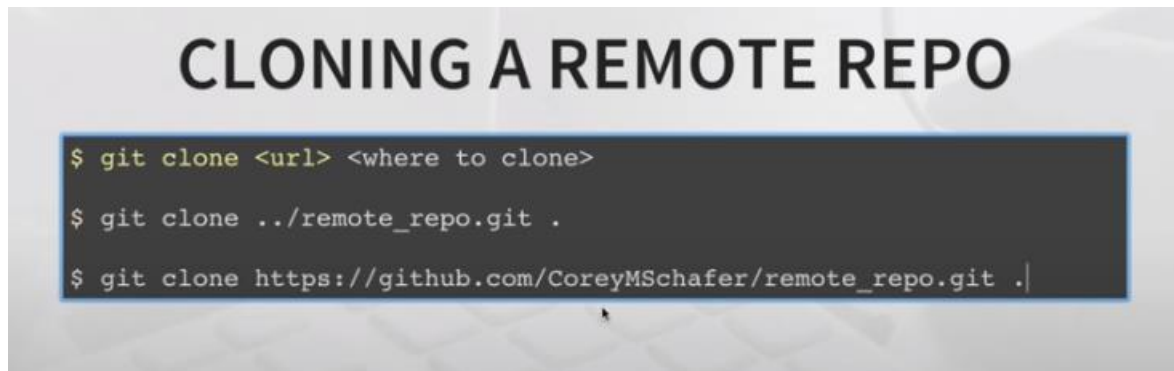
```
$ git reset  
$ git status
```

9. First Commit message

OUR FIRST COMMIT

```
$ git add -A  
$ git commit -m "Initial Commit"  
$ git status  
$ git log
```

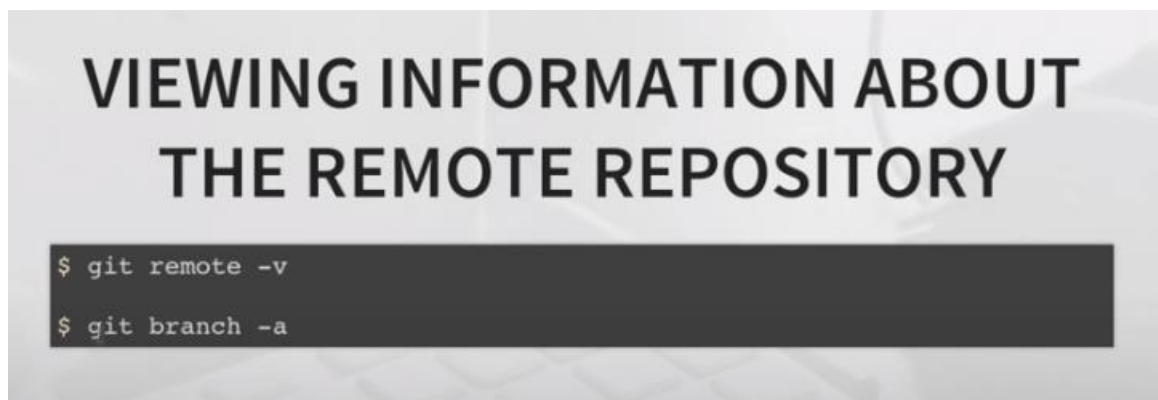
10. Cloning a repo from github (“.” Represents the current working directory)



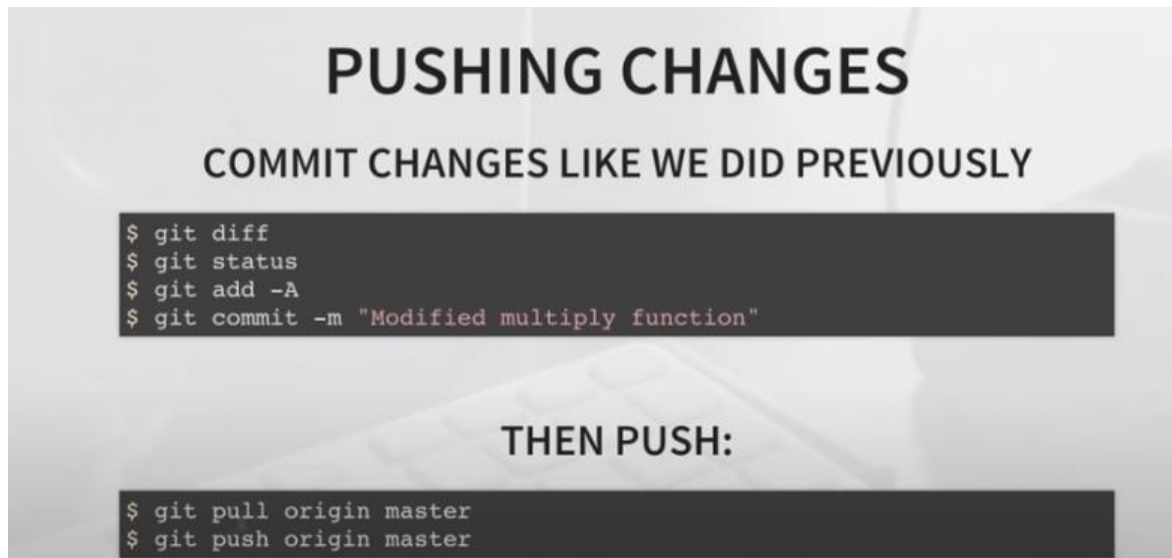
For Example :

\$ git clone <https://github.com/csimgmcet/Cheatsheets.git> .

11. Viewing info about the repository and checking for current branch.

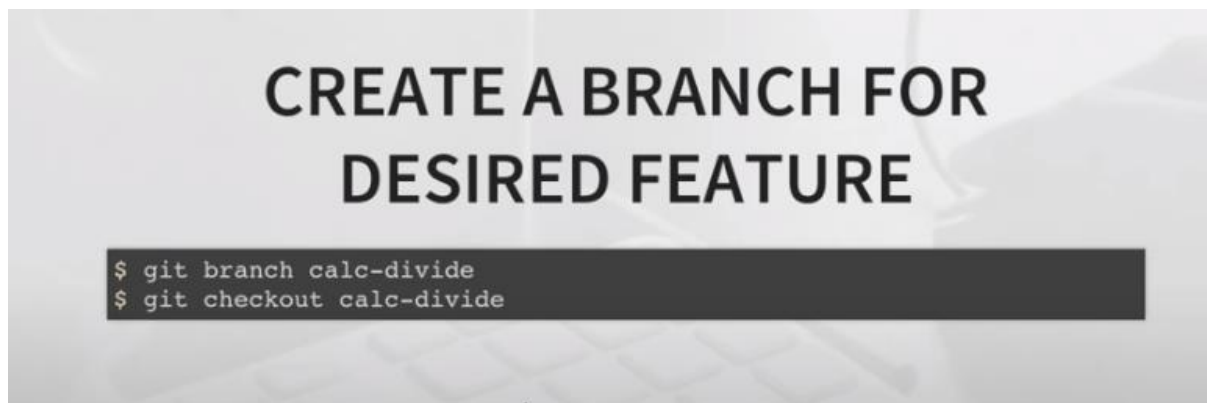


12. Push and pull



If you want to work on a team the best practice is to create multiple branches and then work on it with the work assigned to you.

13. Creating a branch for working on a feature.



Above you were working on a single branch “master”.

Now you will be working on “calc-divide” branch.

```
coreyschafer at Coreys-iMac using -bash in Cloned-Repo on master
$ git branch calc-divide

coreyschafer at Coreys-iMac using -bash in Cloned-Repo on master
$ git branch
  calc-divide
* master

coreyschafer at Coreys-iMac using -bash in Cloned-Repo on master
$ git checkout calc-divide
Switched to branch 'calc-divide'

coreyschafer at Coreys-iMac using -bash in Cloned-Repo on calc-divide
$ git branch
* calc-divide
  master

coreyschafer at Coreys-iMac using -bash in Cloned-Repo on calc-divide
$
```

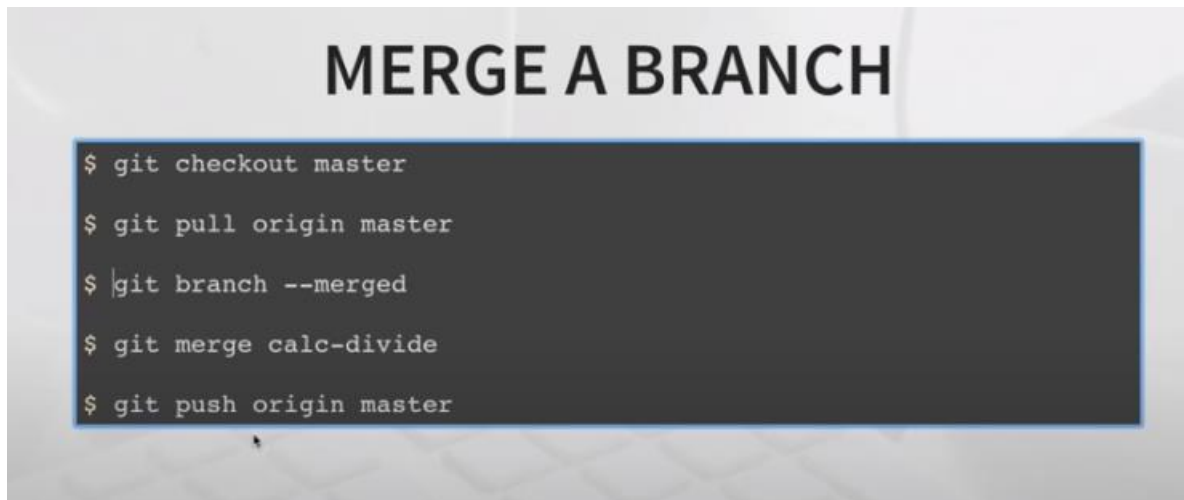
14. Once you have completed making changes in your <your name> branch (here it is calc-divide) you can push the changes to the remote repository.

AFTER COMMIT PUSH BRANCH TO REMOTE

```
$ git push -u origin calc-divide
$ git branch -a
```

git branch -a specifies the number of branches in the terminal.
Here we have 2 (master, calc-divide).

15. Once you are done finalizing your changes you can merge your branch with the main branch (master)



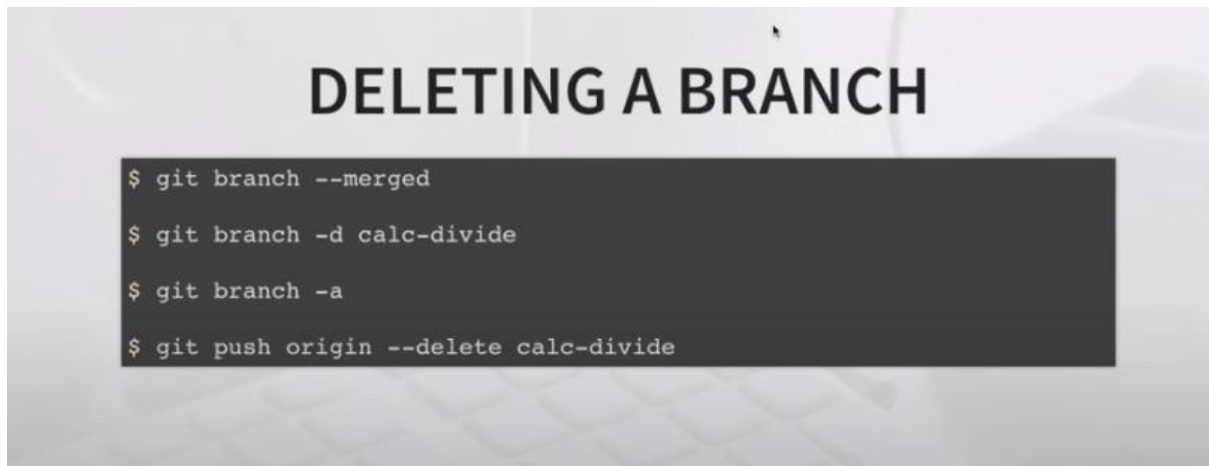
***git checkout master* // to switch to master branch**

***git pull origin master* // to see if someone else have made any changes to the original repo (master)**

***git merge calc-divide* // merges calc-divide with master**

***git push origin master* // pushed changes on master branch of remote repository.**

16. Deleting the calc-divide branch as we have merged it with our main branch (master)



- -d species *delete*.
- Now when you run git branch --merged you will see both the branches.
- But when you run git branch -d calc-divide the calc-divide branch will be deleted.
- Git branch -a will show only one branch (master) .
- To delete calc-divide from our remote repository we use the last command.