Git-Github and Open Source:

Welcome to the World of Freedom.

A Presentation By:

- Ansh Sachdeva
- www.github.com/chaostools
- https://www.linkedin.com/in/anshsachdevaprofessional/

OpenSource:

- What is Open Source?
- Why Open Source?
- How to Open Source?

- What?
- Why?
  - Cloud Backup
  - Show your Journey
  - Helping
  - Getting ‘Helped’ 😜
  - Collaborations.
- How?
  - Git , Github , Gitlab , Mercurial,etc

git
I prefer the real version control
SUBVERSION
I said the *real* version control
Final 2
Final Final
FINAL FOR REAL THIS TIME
OK This is definitely final
Perfection

GitHub

- **History:**
- **Features:**
  - Public/Private Backups.
  - Show Experience and Achievements
  - Helping Others.
  - Getting Helped :p
  - Collaboratives(Open Sourcing)
  - ...Many More
  - Version Control

(https://en.wikipedia.org/wiki/GitHub)

GIT

- **History:**
- **Features:**
  - **Offline VCS tool.**
  - **For Interacting with Online VCS tools like Github , Gitlab , etc.**

Terminologies(github/git)

- Repository(Repo)
- Cloning(`git clone)`
- Forking
- Raising Issues
- Pull Request(P.R)
- Readme and markdown .
- Branches

- `git init`
- `git clone`
- `git ignore`
- `git status`
- `git log`
- `git add .`
- `git commit –m “message”`
- **`git push –u origin master`**
- **`git add origin “url”`**
- **`git pull`**

CODELAB-1 !

1

a) Creating a New repo, clone, add files and push

b) Adding origin to existing project and push

⇒ adding more files to both these project and pushing again

CODELAB-1 !

(in your pc)

1. create an empty folder

(on github.com)

1. create a new repository(// don't initialise with readme)

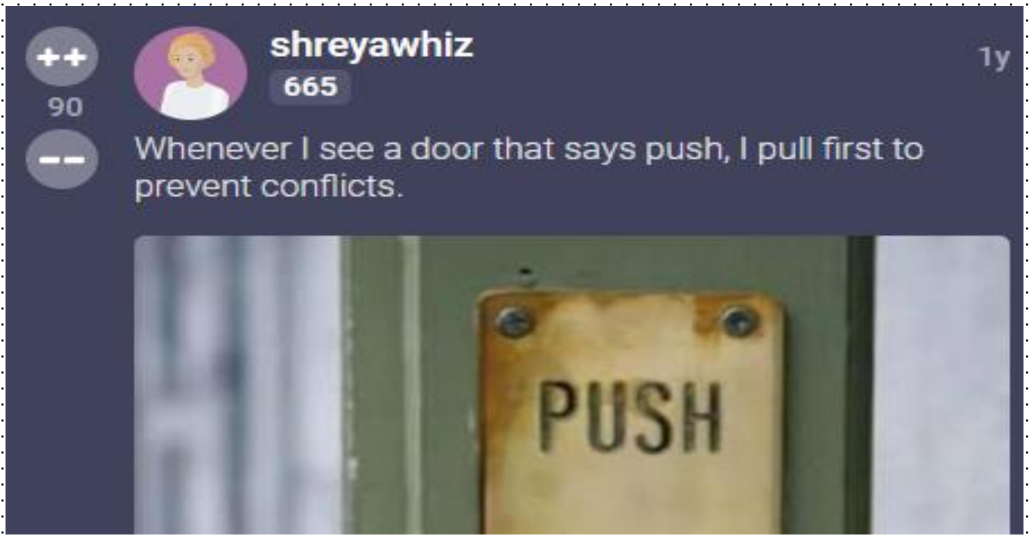
(on terminal)

```
1.  git init
2.  git clone "UrlfromtheGithub'sSite.git"
<<<<,work in this folder(add/modify/edit files)>>>>>
```

(on terminal)

```
1.  git status                               //not necessary
2.  git add .
3.  git commit –m "message"
4.  git status
5.  git push –u origin master
```

Syntaxes



Whenever I see a door that says push, I pull first to prevent conflicts.

```
➢ git push –u origin master
➢ git add .
➢ git init
➢ git status
➢ git clone “URL”
• git log
➢ git commit –m “message”
➢ git remote add origin “URL”
➢ git pull
```

Git Cheat Sheet

REBELLABS by ZEROTURNAROUND

Create a Repository

From scratch -- Create a new local repository
```
$ git init [project name]
```

Download from an existing repository
```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed
```
$ git status
```

Show the changes to files not yet staged
```
$ git diff
```

Show the changes to staged files
```
$ git diff --cached
```

Show all staged and unstaged file changes
```
$ git diff HEAD
```

Show the changes between two commit ids
```
$ git diff commit1 commit2
```

List the change dates and authors for a file
```
$ git blame [file]
```

Show the file changes for a commit id and/or file
```
$ git show [commit]:[file]
```

Show full change history
```
$ git log
```

Show change history for file/directory including diffs
```
$ git log -p [file/directory]
```

Working with Branches

List all local branches
```
$ git branch
```

List all branches, local and remote
```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory
```
$ git checkout my_branch
```

Create a new branch called new_branch
```
$ git branch new_branch
```

Delete the branch called my_branch
```
$ git branch -d my_branch
```

Merge branch_a into branch_b
```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit
```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit
```
$ git add [file]
```

Stage all changed files, ready for commit
```
$ git add .
```

Commit all staged files to versioned history
```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history
```
$git commit -am "commit message"
```

Unstages file, keeping the file changes
```
$ git reset [file]
```

Revert everything to the last commit
```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)
```
$ git fetch
```

Fetch the latest changes from origin and merge
```
$ git pull
```

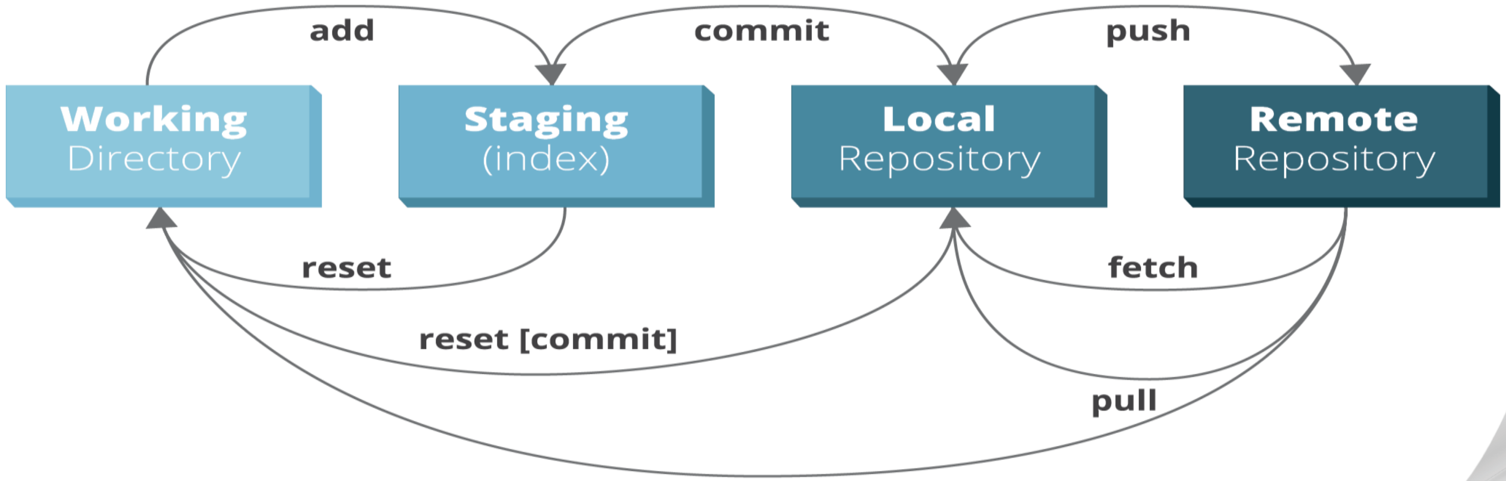
Fetch the latest changes from origin and rebase
```
$ git pull --rebase
```

Push local changes to the origin
```
$ git push
```

Finally!

When in doubt, use git help
```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.



BROUGHT TO YOU BY JRebel

CODELAB-2 !

a) **Pulling the latest changes** :

- Aim:clone CBSession. CB Session will then be changed. pull latest changes

b) Contributing to Open Source Part 1 : **Make a PR**

- *Aim: to add your name in project CB Session*

→fork and clone "CBSession" . Make changes locally and push. Then make a 'PR'

c) Contributing to Open Source Part 2 : **Getting Ready to PR again**

- *Aim: 1) to fetch the latest updates from Project CB Session after Everyone's name has been added.*
  2)Add a short message along with your name about this session(or any other sentence)

→pull changes and **repeat** previous steps(make local changes and push. then make a 'PR'

(on github.com)

1.Fork gitsessions.

------------------------------------Steps from the codelab 1(with a small change)--------------------

(on terminal)

```
1.  git init
2.  git clone "YOUR_FORK's URL"    //(not original git sessions repo)
<<<<,work in this folder(add/modify/edit files)>>>>>
```

(on terminal)

```
1.  git status                                //not necessary
2.  git pull
3.  git add .
4.  git commit -m "message"
5.  git status
6.  git push -u origin master
```

------------------------------------Steps from the codelab 1(with a small change)--------------------

(on github.com)

1.click the 'pull request' button(right of repositories' name)

2. add descriptive info and make a pull request

CODE LAB-2 !