

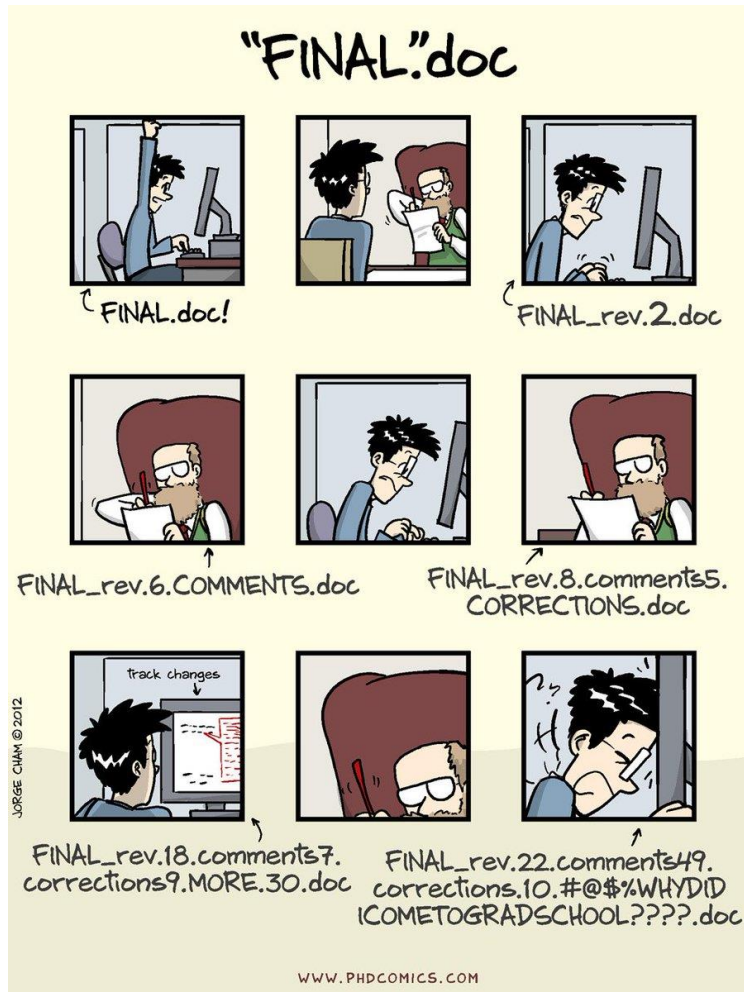
# Just Git it



Prepared by – Sumit Agrawal

# Why git

- Version Control
  - Save work
  - Review Changes
  - Do not lose history
  - Share with others
  - Reduce mental burden
- Working in collaboration



# Installation on Windows

- Git Bash – Command Prompt specifically designed for windows (<https://git-for-windows.github.io/>) {during installation path – use gitbash only}

Youtube video (<https://www.youtube.com/watch?v=tN0T7jYn0A>)

- Configure the git bash (<https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>)

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Username and user email must be same as in used in github.com


# GitHub

- Create an account on github (<https://github.com/>)
- To add an existing project on github follow this. (<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>)
- Public Repository (Free – open)
- Private Repository (create account with iitb mail id – get 2 year of subscription free, after that 7\$ per month)
- Storage(no limitation on number of repository. Repository size<1GB)

Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner  sumitPhD /

Great repository names are short and memorable. Need inspiration? How about friendly-invention.

Description (optional)

---

☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
Only you and those you choose can see and commit to this repository.

---

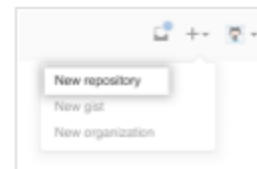
☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

---

[Create repository](#)

- 1 Create a new repository on GitHub. To avoid errors, do not initialize the new repository with *README*, license, or `gitignore` files. You can add these files after your project has been pushed to GitHub.



- 2 Open Git Bash.

- 3 Change the current working directory to your local project.

- 4 Initialize the local directory as a Git repository.

```
$ git init
```

- 5 Add the files in your new local repository. This stages them for the first commit.

```
$ git add .  
# Adds the files in the local repository and stages them for commit. To  
unstage a file, use 'git reset HEAD YOUR-FILE'.
```

- 6 Commit the files that you've staged in your local repository.

```
$ git commit -m "First commit"
# Commits the tracked changes and prepares them to be pushed to a remote
repository. To remove this commit and modify the file, use 'git reset --soft
HEAD~1' and commit and add the file again.
```

- 7 At the top of your GitHub repository's Quick Setup page, click  to copy the remote repository URL.



- 8 In the Command prompt, [add the URL for the remote repository](#) where your local repository will be pushed.

```
$ git remote add origin remote_repository URL
# Sets the new remote
$ git remote -v
# Verifies the new remote URL
```

- 9 [Push the changes](#) in your local repository to GitHub.

```
$ git push origin master
# Pushes the changes in your local repository up to the remote repository you
specified as the origin
```

# EDITOR PAIN

- Download sublime editor
- Make sublime default editor for git bash

# Commands

- `git init`
- `git add abc.m`
- `git status`
- `git commit`
- `git log`
- `git diff abcCommit1 abcCommit2`
- `git remote add origin <url>`
- `git push origin master`
- `git pull`
- `git ignore`



# A note on commit logs

First line brief  $\leq$  50 chars

Detailed information below. Ideally wrapped to 72 cols.

- ALWAYS leave a good log message.
- Bullet points are fine.
- Multiple paras separated by blank line.

# Further Reading

Fetch

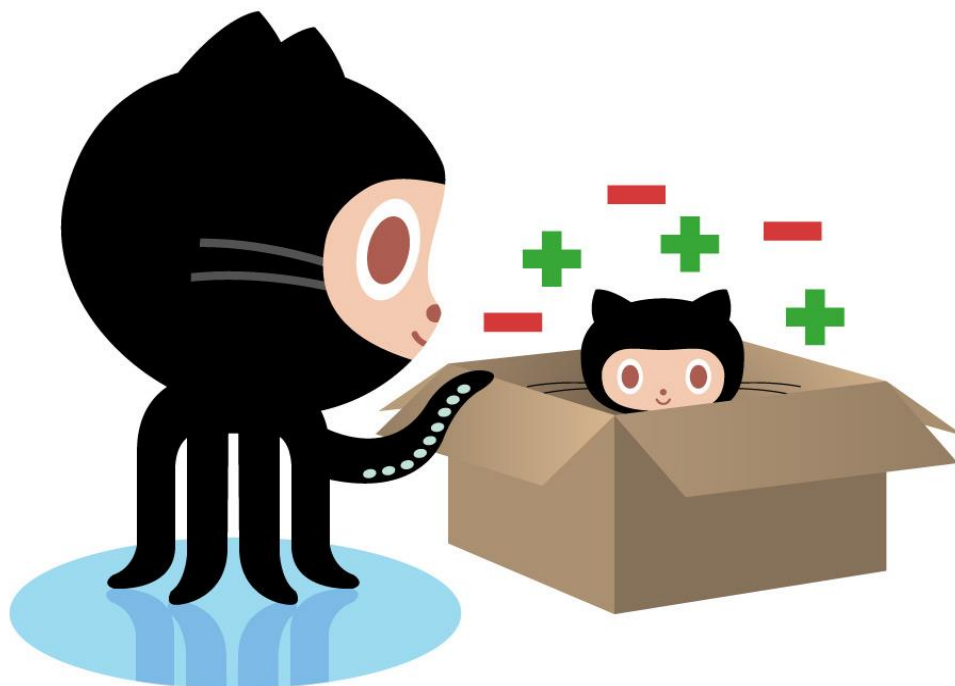
Branching

merge

- Udacity Course – How to use git and github

<https://in.udacity.com/course/how-to-use-git-and-github--ud775/>

# Thank You



happy gitting