

Git: Remotes, Branches & Merges

Prashant K. Sharma
Diptesh Kanojia

e-Yantra Team
ERTS Lab, IIT Bombay

IIT Bombay
April 17, 2021



Agenda for Discussion

1 Git: Recap

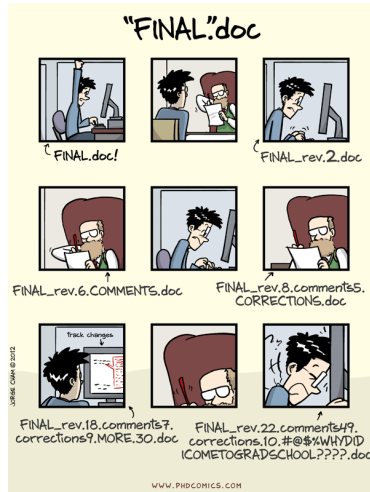
- Why Version Control ?
- Git Overview

2 Git: How ?

- Remote in Git
- Remote repository on GitHub
- Adding changes to remote repository
- Branches
- Merging and Conflicts



Why Version Control ?



Remote in Git

To collaborate with other people, we need to be able to share changes we are doing in our local repository.



Remote in Git

To collaborate with other people, we need to be able to share changes we are doing in our local repository.

Using Git, we can move our local repository over the web. Most developers use, any of the following hosting services :

- GitHub (*Most widely used*)
- GitLab
- BitBucket



Creating a repository on GitHub

To be able to share changes we've made locally. Log in to GitHub, then click on the icon in the top right corner to create a new repository called **super**.



Creating a repository on GitHub

To be able to share changes we've made locally. Log in to GitHub, then click on the icon in the top right corner to create a new repository called **super**.

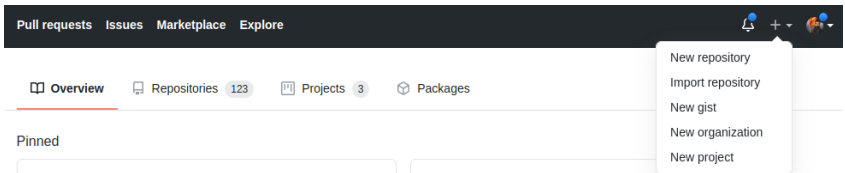


Figure 2: Creating new repository on GitHub

Creating a repository on Github

After naming your repository, adding description etc. click **Create Repository**



Creating a repository on Github

After naming your repository, adding description etc. click **Create Repository**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

 prashantksharma

Repository name *

super

Great repository names are short and memorable. Need inspiration? How about [furry-octo-adventure?](#)

Description (optional)

Test repo for Eyantra Software Labs Course

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None

Add a license: None



Create repository

Figure 3: Adding details for New repository



Creating a repository on Github

After your repository on GitHub, you should be able to see information on how to configure your local repository.



Creating a repository on Github

After your repository on GitHub, you should be able to see information on how to configure your local repository.

The screenshot shows the GitHub repository page for 'prashantksharma/super'. The repository has 1 watch, 0 stars, and 0 forks. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Quick setup' section provides instructions for cloning the repository using HTTPS or SSH, and for creating a new repository or pushing an existing one from the command line.

Quick setup — if you've done this kind of thing before

or **HTTPS** **SSH** `https://github.com/prashantksharma/super.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# super" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/prashantksharma/super.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/prashantksharma/super.git
git push -u origin master
```

Figure 4: GitHub Quick Setup information



Adding changes to remote repository

How do we add remote repository on GitHub ?



Adding changes to remote repository

How do we add remote repository on GitHub ?

Use of **git remote** command

```
$ git remote add origin https://github.com/john/super.git
```

NOTE: replace "john" by your own GitHub username.



Adding changes to remote repository

How do we add remote repository on GitHub ?

Use of **git remote** command

```
$ git remote add origin https://github.com/john/super.git
```

NOTE: replace "john" by your own GitHub username.

How do we verify that remote repository has been added ?



Adding changes to remote repository

How do we add remote repository on GitHub ?

Use of **git remote** command

```
$ git remote add origin https://github.com/john/super.git
```

NOTE: replace "john" by your own GitHub username.

How do we verify that remote repository has been added ?

Use of **git remote** command

```
$ git remote -v
```



Adding changes to remote repository

How do we push all the changes from our local repository to remote repository on GitHub ?



Adding changes to remote repository

How do we push all the changes from our local repository to remote repository on GitHub ?

Use of **git push** command

```
$ git push origin master
```



What is branch ?

The concept of branches in Git, allow us to diverge from our current development and try something new which won't alter the history of our main work. E.g. When working on code bases, we might want to create a new code feature at the same time leave our main code intact.



What is branch ?

The concept of branches in Git, allow us to diverge from our current development and try something new which won't alter the history of our main work. E.g. When working on code bases, we might want to create a new code feature at the same time leave our main code intact.

How do we create a new branch called "experiment" for our repository i.e "super"?



What is branch ?

The concept of branches in Git, allow us to diverge from our current development and try something new which won't alter the history of our main work. E.g. When working on code bases, we might want to create a new code feature at the same time leave our main code intact.

How do we create a new branch called "experiment" for our repository i.e "super"?

Use of **git branch** command

```
$ git branch experiment
```



Switching branch

How do we switch to the new branch called "experiment"?



Switching branch

How do we switch to the new branch called "experiment"?

Use of **git checkout** command

```
$ git checkout experiment
```



Switching branch

How do we switch to the new branch called "experiment"?

Use of **git checkout** command

```
$ git checkout experiment
```

How do we create and switch to the branch using a single command ?



Switching branch

How do we switch to the new branch called "experiment"?

Use of **git checkout** command

```
$ git checkout experiment
```

How do we create and switch to the branch using a single command ?

Use of **git checkout** command

```
$ git checkout -b experiment
```



Merging

As of now, we have two branches namely: **master** and **experiment**. Let us assume that we are working on **experiment** branch. Making changes and committing changes.



Merging

As of now, we have two branches namely: **master** and **experiment**. Let us assume that we are working on **experiment** branch. Making changes and committing changes.

How do we merge the changes done on **experiment** branch to **master** branch ?



Merging

As of now, we have two branches namely: **master** and **experiment**. Let us assume that we are working on **experiment** branch. Making changes and committing changes.

How do we merge the changes done on **experiment** branch to **master** branch ?

Use of **git merge** command

```
$ git checkout master
```

```
$ git merge experiment
```



Merging

How do we delete the **experiment** branch after merging the changes ?



Merging

How do we delete the **experiment** branch after merging the changes ?

Use of **git branch** command

```
$ git branch -d experiment
```



Merging

How do we delete the **experiment** branch after merging the changes ?



Merging

How do we delete the **experiment** branch after merging the changes ?

Use of **git branch** command

```
$ git branch -d experiment
```



Conflicts

Most of the time, doing **git merge** will work smoothly. Unless, two branches are in conflict: e.g. due to changes made to the same file etc.



Conflicts

Most of the time, doing **git merge** will work smoothly. Unless, two branches are in conflict: e.g. due to changes made to the same file etc.

As of now, we have two branches namely: **master** and **experiment**. Let us assume that we encounter a conflict due to changes made to a file `test.txt` which is present on both **experiment** branch and **master** branch.



Conflicts

How to resolve merge conflict ?

Key steps to be taken

- `git merge experiment`
- Open `test.txt` in an editor and keep relevant changes.
- `git commit -a`



References

- Git Cheat Sheet
- Visualizing Git



Thank You!

Author: Prashant K. Sharma
Contributor: Diptesh Kanojia

Post your queries at: resources@e-yantra.org

