

Github and Git

Sarith Divakar M

Agenda

- Version control
- Git
- Basic Git Operations
- Github
- Important Concepts for Github Users
- Hands-on session for Research Scholars
- Github Markdown for Research Documentation
- Github Education

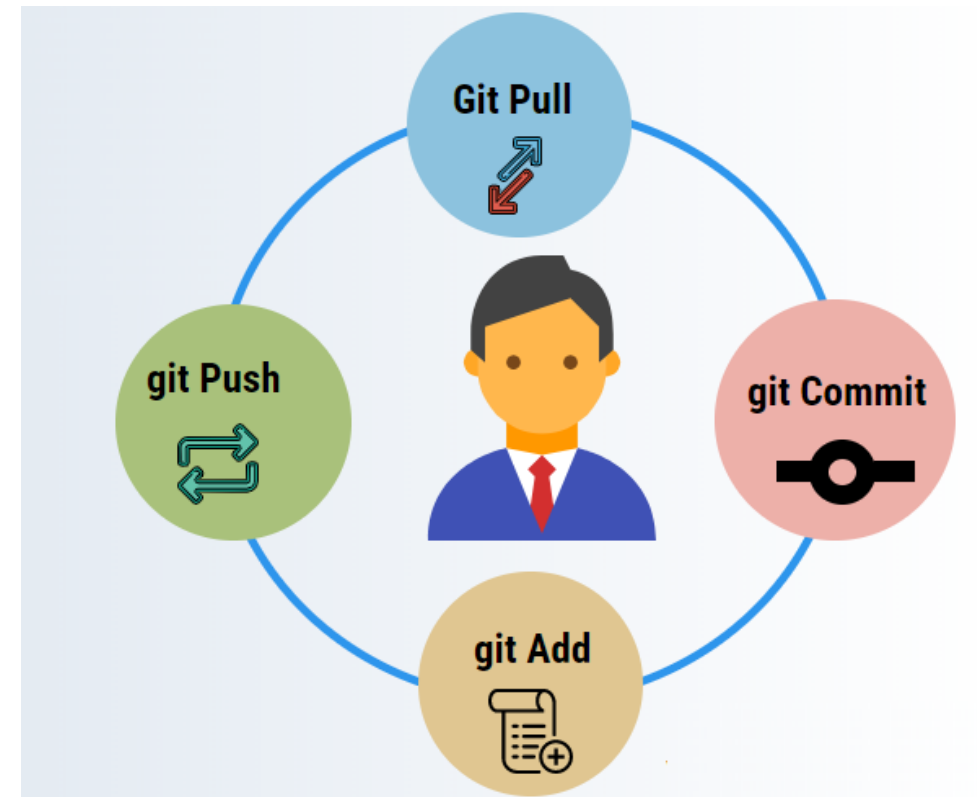
Version control

- Version control is a system that records changes to a file or set of files and helps us recall specific versions later if needed
- It allows you to :
 - Revert files or the whole project to an earlier state
 - Compare changes over time
 - See who modified what?
 - Control modifications by collaborators

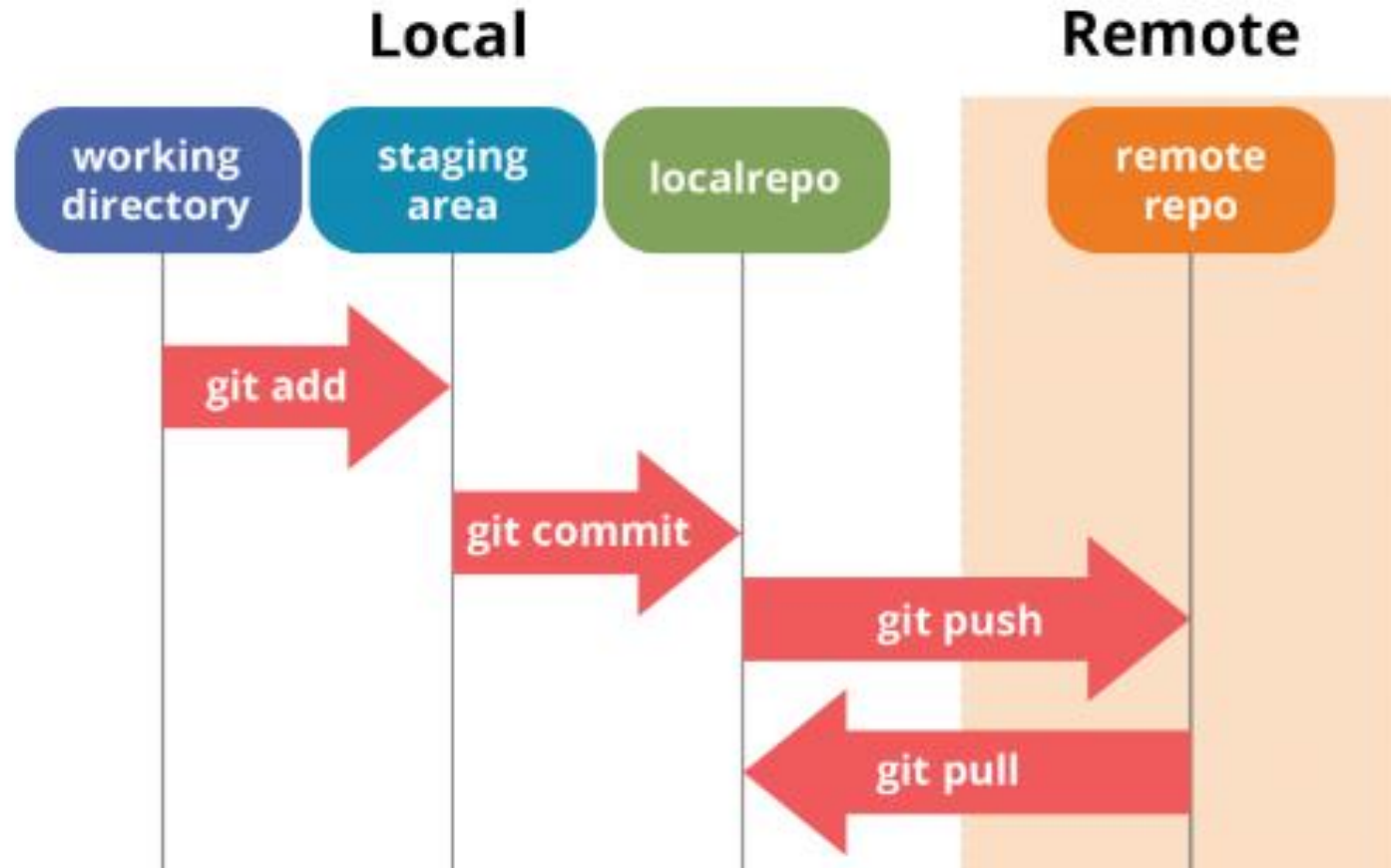
Git



- Git is an example of version control system created by Linus Torvald
- You don't need network for
 - Performing a diff
 - Viewing file history
 - Committing changes
 - Merging branches
 - Obtaining any other revision of a file
 - Switching branches

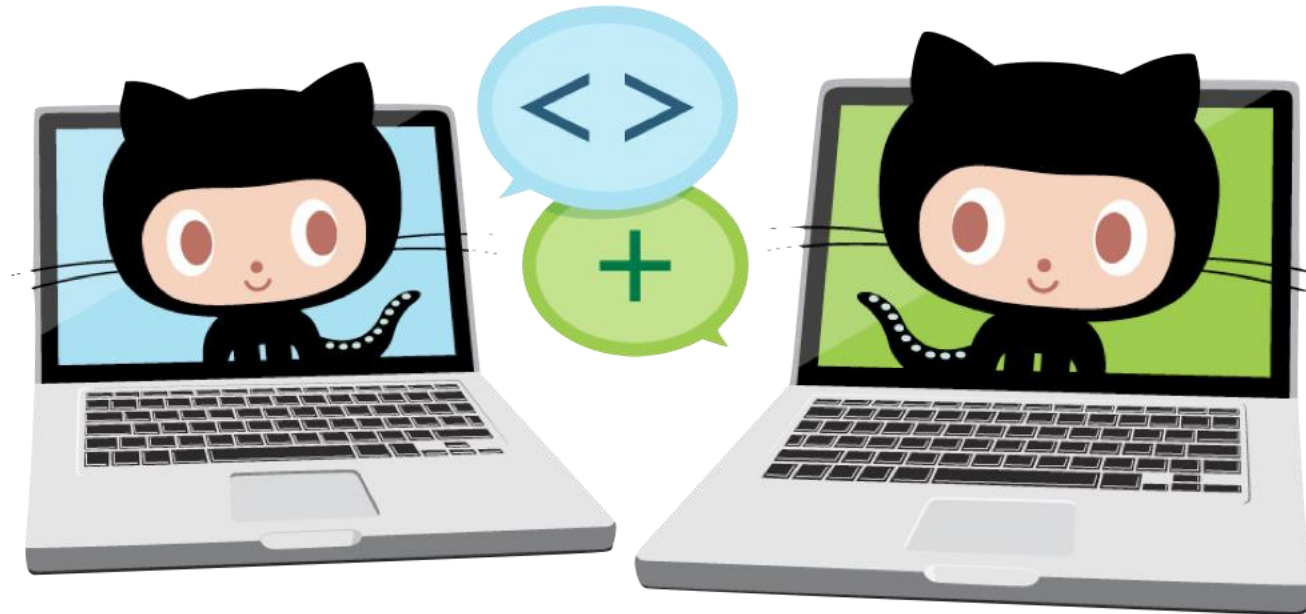


Basic Git Operations





- **Github** is a repository hosting service for Git
- GitHub is where over 56 million developers shape the future of software, together



Important Concepts for Github Users

- **Creating a repo**
 - Creating a repository for multiple people to work together
- **Forking a repository**
 - It creates a copy for you to work on independently without any changes to theirs.
 - Submit a pull request to owner so that the owner can incorporate changes.
- **Adding Commits**
 - Keeps track of your progress as you work on a branch or master.
 - Creates a transparent history that others can follow to understand what you've done and why.

Important Concepts for Github Users

- **Master in a repository**

- This is the final version that is considered ready to use by anybody in the team or outside if repository is public.

- **Creating a Branch**

- Create a branch in your project, for an environment where you can try out new ideas.
- Changes you make on a branch don't affect the master unless pull request is accepted.
- Changes committed to branch reflects for you to keep track of different versions

Important Concepts for Github Users

- **Pull requests**

- Pull Requests initiates discussion about your commits or changes made to a code.
- See exactly what changes would be merged if pull request is accepted.
- Use GitHub's @mention system in your Pull Request message to ask for feedback from specific people or teams, or for someone to review your work

- **Issues**

- Highlight bugs or issues with codes that need rectification.
- Issues remain open unless resolved.
- Can be filtered, Can be labeled as bug/enancement/ question/help wanted etc
- @mention can be used to notify someone

Let's Practice

- Create a “repository” (project) with a git hosting tool (like Github)
- Copy (or clone) the repository to your local machine
- Add a file to your local repo and “commit” (save) the changes
- “Push” your changes to your master branch
- Make a change to your file with a git hosting tool and commit
- “Pull” the changes to your local machine
- Create a “branch” (version), make a change, commit the change
- Open a “pull request”.
- “Merge” your branch to the master branch

Hands-on

- Linux Users/Linux users with Windows Subsystem for Linux
- Windows Users with Visual Studio Code
- Git Version Control with Jupyter Notebooks
- <https://github.com/sarithdm/gitandgithub/blob/main/README.md>

Markdown

- <https://github.com/sarithdm/gitandgithub/blob/main/schedule.md>
- <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>



<https://education.github.com> Note: Use @cusat.ac.in



GitHub Student Developer Pack

The best developer tools, free for students



GitHub Campus Experts

Training to enrich the technology community at your school



GitHub Teacher Toolbox

The best developer tools for teaching, free for academic use



GitHub Campus Advisors

Teacher training to master Git and GitHub



GitHub Classroom

The best way for teachers to distribute and collect coursework on GitHub



GitHub Campus Program

GitHub for your whole school, with everything you need to make it great