GIT AND GITHUB

What is Git ???

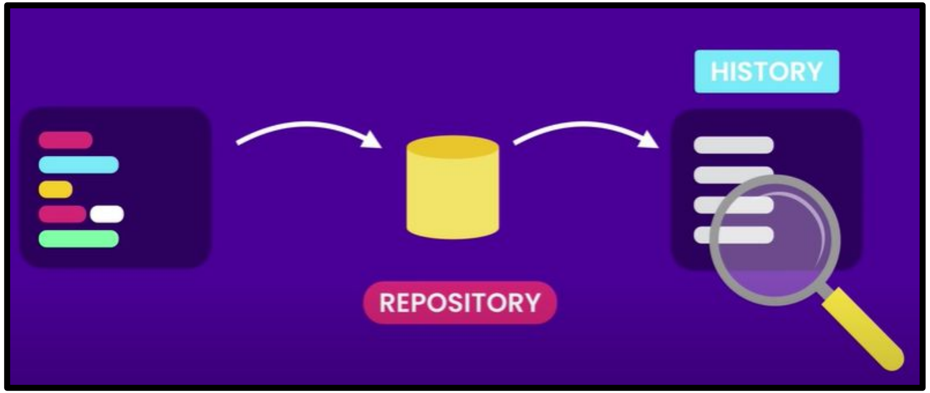
- It is one of the most popular ***Version Control System***.

- A version control system records the change made to our code over time in a special database called repository.

- The changes so made are listed down in a log file along with all the necessary data.

- As a result we can look into this log file at any point of time and see who has made what changes, when and why.

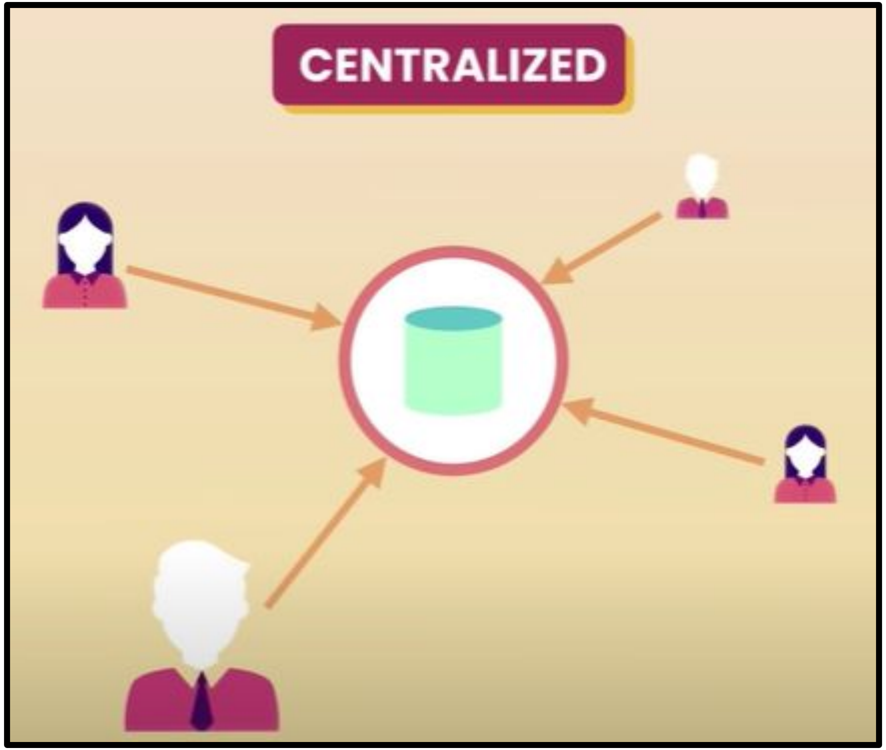
- At any point of time if something goes wrong, it offers the feature to revert back to previous state.



Types of Version Control System

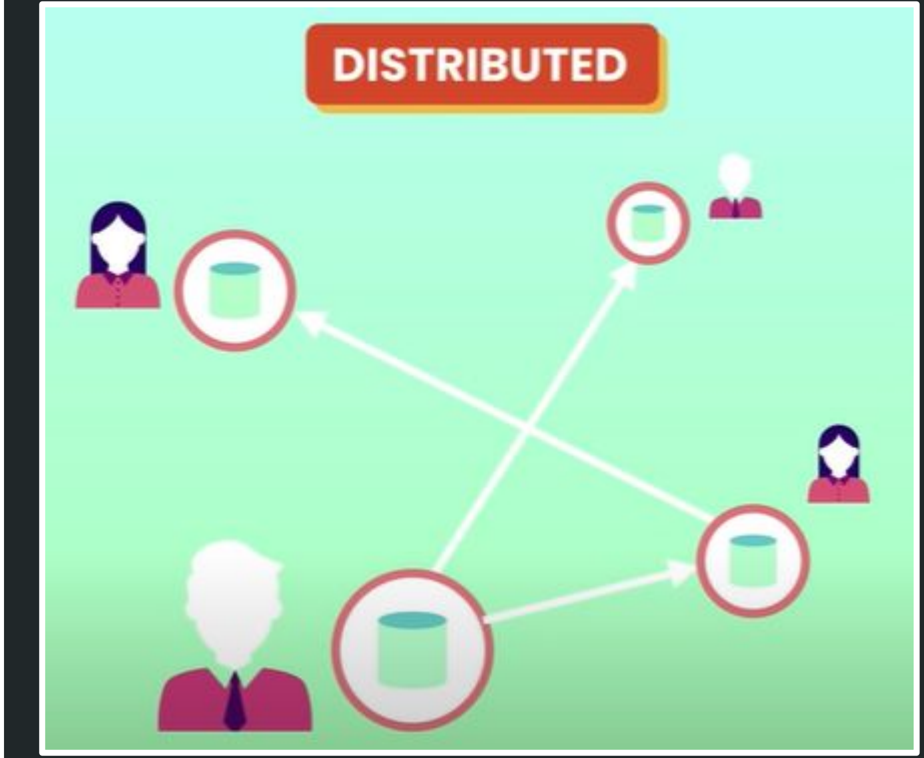
1. ***Centralized VCS*** -

- Centralized version control systems contain just one repository and each user gets their own working copy.
- You need to commit to reflecting your changes in the repository.
- It is possible for others to see your changes by updating.
- Two things are required to make your changes visible to others which are:
    - ❏ You commit
    - ❏ They update
- Eg -
    - ❏ Subversion
    - ❏ Team Foundation Server



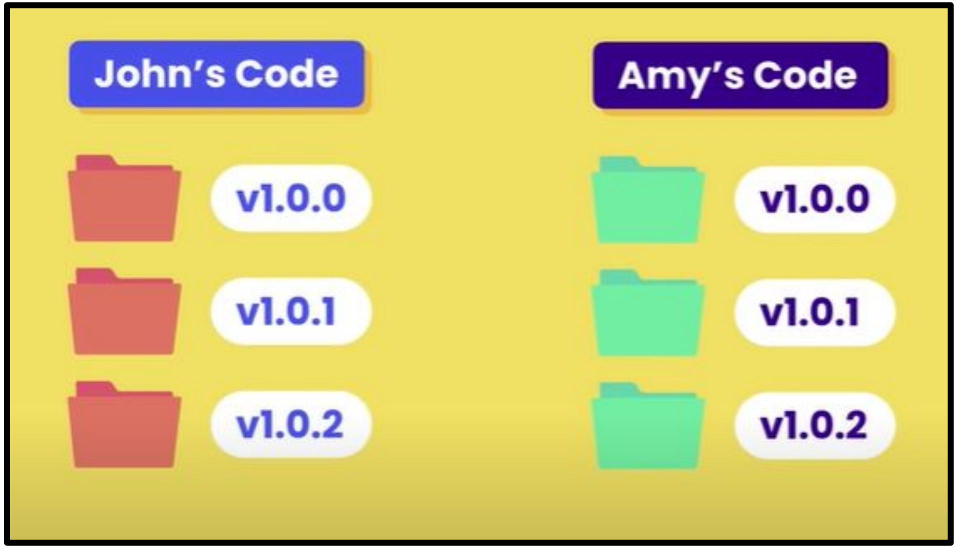
2. ***Distributed VCS*** -

- Distributed version control systems contain multiple repositories.
- Each user has their own repository and working copy. Just committing your changes will not give others access to your changes.
- This is because commit will reflect those changes in your local repository and you need to push them in order to make them visible on the central repository.
- To make your changes visible to others, 4 things are required:
    - You commit
    - You push
    - They pull
    - They update
- Eg -
    - Git
    - Mercurial



Why Git ???

- No centralization
- Cloning creates a replica on local device
- Easily track state, history and integrity
- Branching and merging becomes easier
- Fast (only one .git repository)
- Files to be committed are “**staged”** first
- Free and open source
- Flexible Workflow



What is Github ???

- GitHub is a for-profit company that offers a cloud-based Git repository hosting service.
- Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.
- GitHub’s interface is user-friendly enough so even novice coders can take advantage of Git.
- Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.
- Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

Github Essentials

1. ***Repositories*** -
   - A GitHub repository can be used to store a development project.
   - It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images).

2. ***Branches*** -
   - A GitHub branch is used to work with different versions of a repository at the same time.
   - By default a repository has a master branch (a production branch).

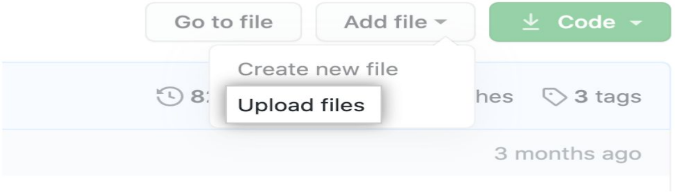
3. ***Commits*** -
   - At GitHub, changes are called commits.
   - Each commit (change) has a description explaining why a change was made.

4. ***Pull Requests*** -
   - Pull Requests are the heart of GitHub collaboration.
   - With a pull request you are proposing that your changes should be merged (pulled in) with the master.

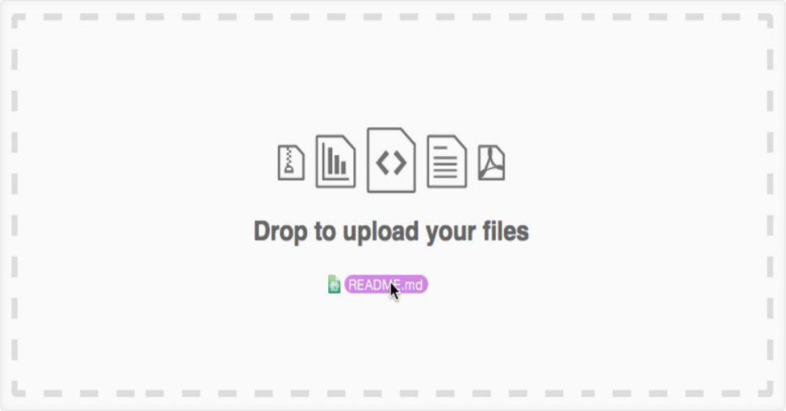
Methods to upload Project on Github

A) ***Using Github Platform*** -

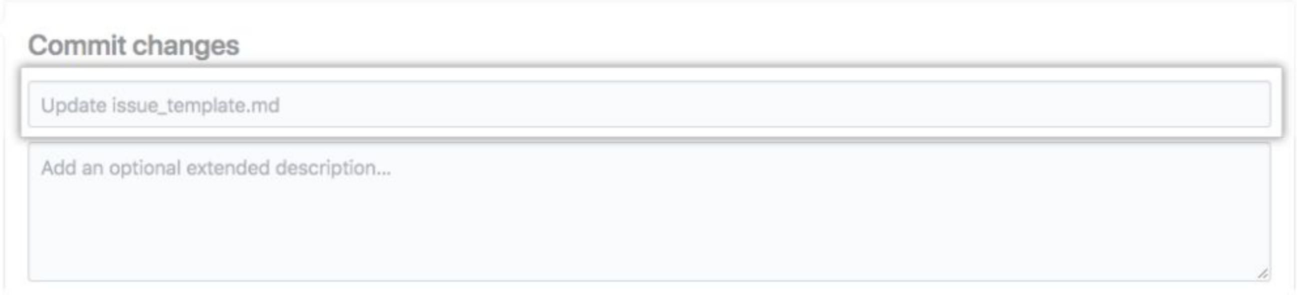
1. On GitHub, navigate to the main page of the repository.
2. Above the list of files, using the Add file drop-down, click Upload files.



3. Drag and drop the file or folder you'd like to upload to your repository onto the file tree.



4. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file.

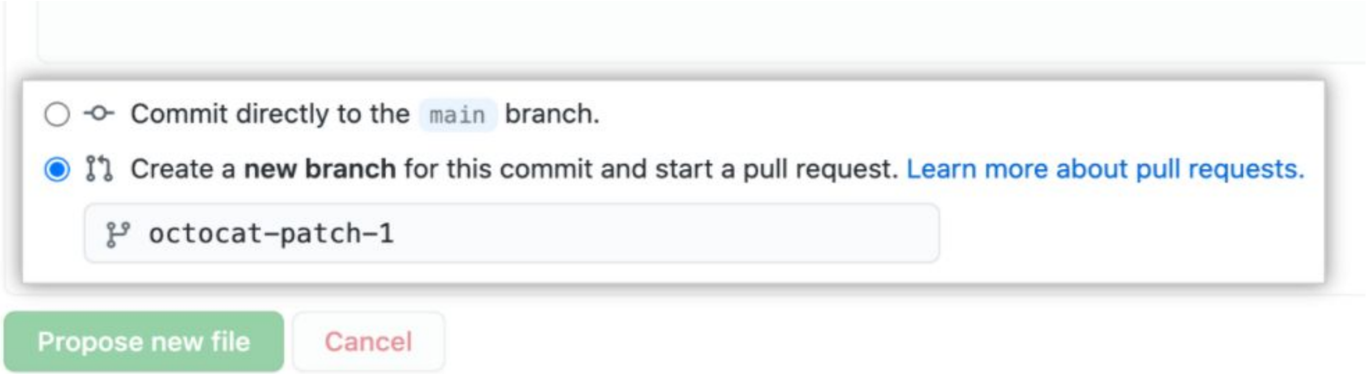


Commit changes

Update issue_template.md

Add an optional extended description...

5. Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is the default branch, you should choose to create a new branch for your commit and then create a pull request.

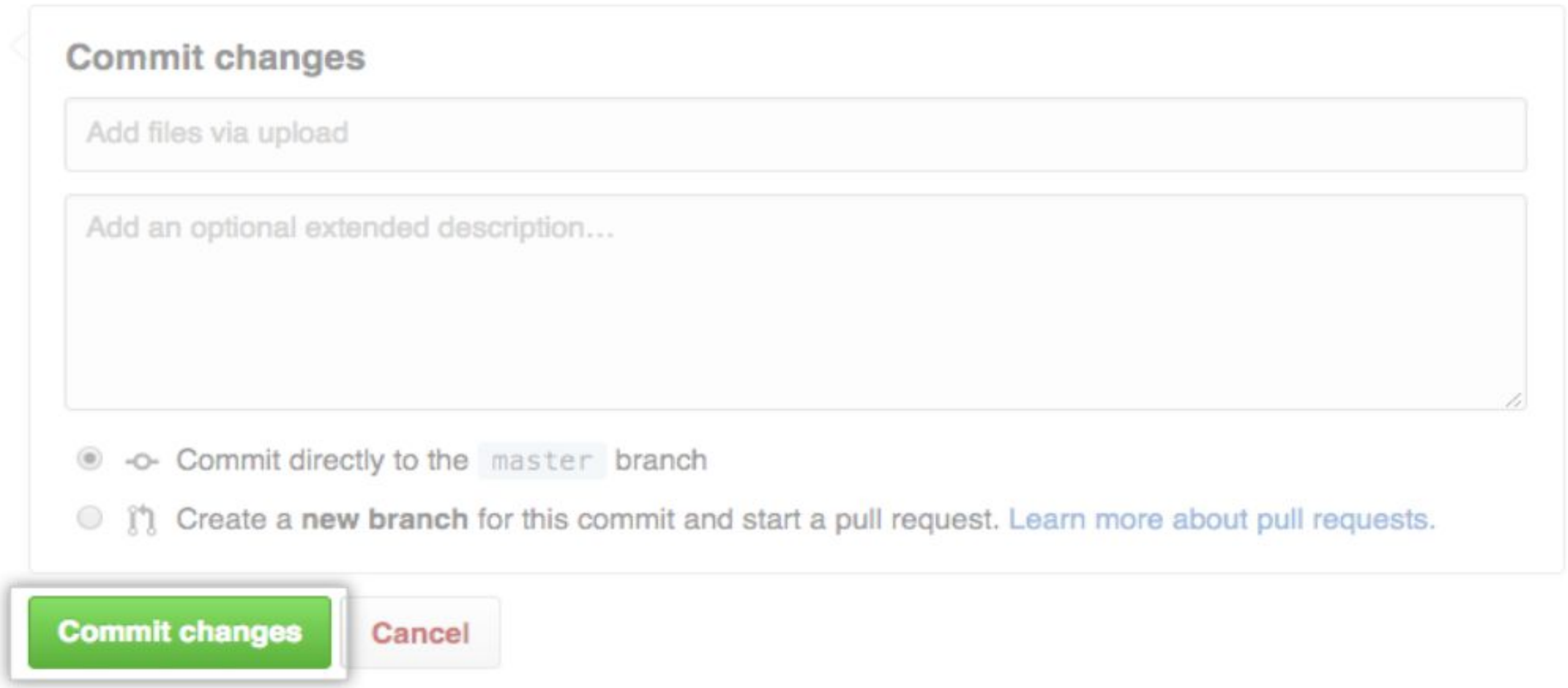


○ Commit directly to the `main` branch.

● Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

octocat-patch-1

6. Click Commit changes.



Commit changes

Add files via upload

Add an optional extended description…

● Commit directly to the `master` branch

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

B) ***Using Command Line*** -

$ git config :

This command sets the author name and email address respectively to be used with your commits.

```
Syntax : git config --global user.name “[name]”
                git config --global user.name "abc”
Syntax: git config --global user.email “[email address]”
                git config --global user.emali "abc@gmail.com"
```

$ git init :

The git init command initialize a new, empty Git repository.It generates a file as ".git" which is by default hidden.

```
Syntax: git init
```

$ git add :

The git add command adds a change in the working directory to the staging area.

```
Syntax : git add . (add all files in the folder)

                git add [filename] (adds single file)
```

$ git commit :

In Git, commit is the term used for saving changes.
Git does not add changes to a commit automatically.
You need to indicate which file and changes need to be saved before running the Git commit command.

Syntax : git commit -m “[Type in the commit message]”

$ git remote :

This command is used to connect your local repository to the remote repository.

Syntax : git remote add [variable name] [Remote Server Link]

$ git pull :

This command fetches and merges changes from a remote repository and immediately update the local repository

Syntax : git pull [variable name] [branch name]

$ git push :

The git push command is used to upload local repository content to a remote repository.

Syntax : git push [variable name] [branch name]

THANK YOU