5_git

What is Git?

- **Definition**: Git is a tool that helps track changes in code and allows multiple people to work on the same project without messing up each other's work.
- **Key Features**:
  - **Version Control**: Keeps a history of all changes made to the project.
  - **Distributed**: Every developer has a full copy of the project.
  - **Branching and Merging**: Allows working on different tasks simultaneously and combining changes easily.
  - **Efficiency**: Handles large projects well.

Benefits of Using Git

- **Collaboration**: Multiple people can work on the same project at the same time.
- **History**: Keeps track of all changes, making it easier to fix bugs and understand the project's development.
- **Branching**: Work on different features or fixes without affecting the main project.
- **Backup**: Each copy of the project is a complete backup.
- **Integration**: Works well with GitHub, GitLab, and Bitbucket for remote collaboration

installation

```
* windows : https://git-scm.com/download/win
* macos: https://sourceforge.net/projects/git-osx-installer/
* linux : https://www.git-scm.com/download/linux
```

Setup

- check whether you git is installed or not

```
git --version
```

config addition -> name , email

- list of configs

```
git config --list
```

- add your username and your email id

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

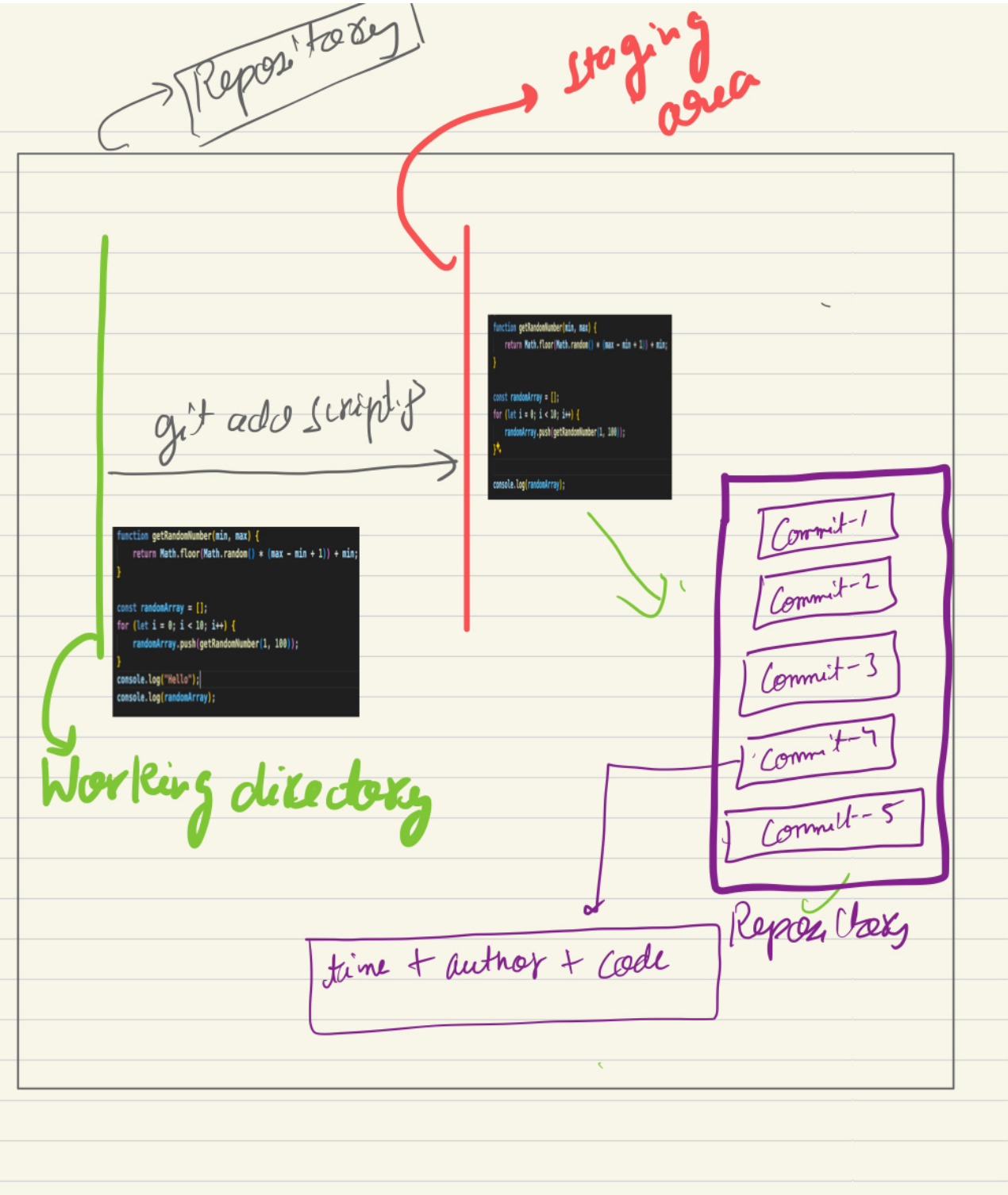
How git stores the changes

- git has three areas -> working directory-> current changes
- Intermediate layer -> staging
- When developer things that changes are finalised then it is `commited` to repository

Terms

1. **Repository (Repo):**
   A storage location for your project, including all the files and the entire history of their changes.
2. **Stage (or Index)**:
   - The area where changes are prepared before being committed. You add files to the staging area using `git add`.
3. **Commit**:
   - A snapshot of changes made to the repository. Each commit has a unique ID and a message describing the changes.
4. **Remote**:
   - A version of the repository that is hosted on a server, such as GitHub. Remote repositories enable collaboration by allowing multiple people to work on the same project.
5. **Push**:

- Sending your local commits to a remote repository. This updates the remote repository with your changes.
  staging area -> tracked changes
  working directory -> untracked changes



git workflow

- git init -> wherever you enter git init an empty git repository is created (tracker without any history)
- `git add` -> added to staging area(snapshot of the code is added)
- `.gitignore` : this the file where you can put the files and folder name which you don't want to be tracked
- `git add .` -> to send all the files current snapshot to staging area
- `git status` -> differenece between staging and working directory
- `git commit -m "commit message"` : A snapshot of changes made to the repository. Each commit has a unique ID and a message describing the changes.
- `git log` : list of commits
- `git checkout commit-hash` : you can view how code looked like in a particular commit
    - to go back to latest commit -> `HEAD`
        - type git branch -> check your branch name
        - `git checkout branch-name`

commands to push your changes

Setup remote repo

- create a repo in github
- git remote add origin `remote-repo-name`
- git branch -M main
- git push -u origin main